

Design • Build • Program • Learn • Compete

SERVO

FOR THE ROBOT INNOVATOR

www.servomagazine.com

MAGAZINE

August 2013

Making Better Arduino Robots ARD BOT II

With The



◆ **PR2Lite Grows Up**
The latest evolution of this homebrewed PR2.

◆ **It's All In Your Head**
How scientists are looking to the anatomy of the human body to make smarter and faster machines.

MAKE YOUR ROBOT ROCK!



Get ready to rock with Hitec's
ultra heavy duty monster servos!

Designed with all aluminum EMI-shielded cases, knock-out steel gears and non-programmable digital circuits, the industrial strength HS-900SGS and HS-1000SGT deliver the power to make your robot a rock star!



	11.1 Volts		14.8 Volts	
	Speed	Torque	Speed	Torque
HS-900SGS	0.20 sec/60°	903 oz-in	0.15 sec/60°	1208 oz-in
HS-1000SGT	0.26 sec/60°	1167 oz-in	0.19 sec/60°	1528 oz-in
Dimensions	2.52 x 1.30 x 2.87 in		Weight	12.80 oz

For those about to ROBOT, we salute you!



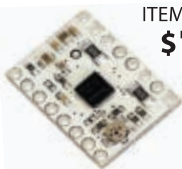


38 kHz IR Proximity Sensor

ITEM #2460

\$5⁹⁵

- Typical sensing range up to 24" (60 cm)
- Fixed-gain modulated IR detector
- Small size (0.4" × 0.6")



ITEM #2134

\$7⁹⁵

(qty. 5)

- DRV8834 driver
- 4-layer PCB
- 1/32 microstepping
- Low voltage (2.5 V - 10.8 V)

Stepper Motor Drivers

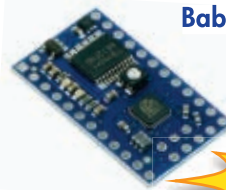


ITEM #1182

\$7⁹⁵

(qty. 5)

- A4988 driver
- 1/16 microstepping



Baby Orangutan B-328 Robot Controller

ITEM #1220

\$19⁹⁵

Complete control solution for small robots

- Integrated dual motor drivers
- User-programmable ATmega328P AVR
- Small 24-pin DIP form factor (1.2" × 0.7")



170-Point Breadboard

ITEM #1488

\$2⁹⁵

Other colors available

- 17 rows of tie points for up to two 14- or 16-pin DIP ICs
- Tabs allow multiple units to be connected for larger projects



Invenscience i00600 Torxis Servo

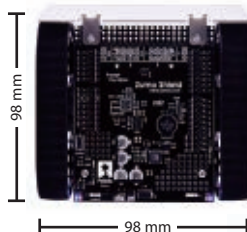
ITEM #1390

\$289

@12V:

- 1 sec/60°
- 1600 oz·in

Ultra-high-torque monster servo



Zumo Robot for Arduino (Assembled with 75:1 HP Motors)

ITEM #2506

\$99⁹⁵

Build and customize your own Zumo!

Arduino-controllable tracked robot small enough for mini-sumo (less than 10 cm × 10 cm) and flexible enough for you to make it your own. If you build it, it will push!



NEMA 23 Hybrid Stepper Motor

ITEM #1477

49⁹⁵

Other sizes available

- 1 A/phase @ 8.6 V
- 1.8° step angle
- 190 oz·in
- 56.4 mm × 56.4 mm × 76 mm



Ball Caster with 3/4" Metal Ball

ITEM #955

\$2⁹⁹

Other sizes available

Our ball casters make a great third contact point for small two-wheeled robots.



Dual MC33926 Motor Driver Shield for Arduino

ITEM #2503

\$29⁹⁵

Control two brushed DC motors with an Arduino or Arduino-compatible board. Operates between 5 V and 28 V and can deliver a continuous 3 A per motor.

Step-Up/Step-Down Voltage Regulator S10V4F5

ITEM #2121

\$4⁴⁹

9V and 12V versions available

- 2.5 V to 18 V input
- Fixed 5 V output
- 70% to 80% efficiency



Universal Aluminum Mounting Hub Pair

ITEM #1079

\$5⁹⁵/pair

Other shaft sizes available

Mount custom wheels and mechanisms to 3 mm motor shafts.



3pi Robot

ITEM #975

\$99⁹⁵

The 3pi is a great first robot for ambitious beginners and a perfect second robot for those looking to move up from non-programmable or slower beginner robots.



Simple Motor Controller 18v15

ITEM #1377

\$39⁹⁵

Advanced motor control made simple

Highly configurable DC motor controller that supports four interface modes: USB, TTL serial, analog voltage, and hobby radio control (RC).



30-Tooth Silicone Track Set

ITEM #1416

\$14⁹⁵

Available with shorter tracks

Track set includes a pair of 30-tooth silicone tracks, two drive sprockets measuring 1.38" (35 mm) in diameter, and two matching idler sprockets along with mounting hardware.



Mini Plastic Gearmotor

ITEM #1125

\$5⁴⁹

Other shapes and gear ratios available

@4.5V:

- 120 RPM
- 20 oz·in



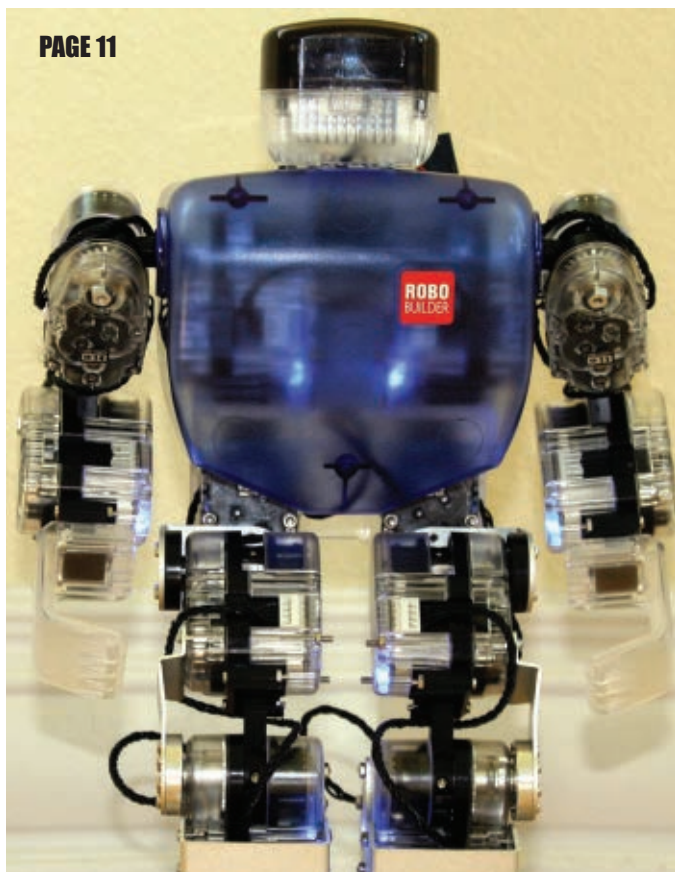
25D mm Metal Gearmotors with 48 CPR Encoders

\$36⁹⁵

Other sizes available

Many motor winding and gear ratio combinations stocked.

PAGE 11



08 Robytes

by Jeff Eckert

Stimulating Robot Tidbits

- Robo Suit in Final Trials
- Insect Vision for Bots and More
- Aerobatic MAV Takes Flight
- Bot Flips the Bird
- Sniff This!

11 Ask Mr. Roboto

by Dennis Clark

Your Problems Solved Here

Putting the RoboBuilder Creative kit in motion.

70 Twin Tweaks

by Bryce and Evan Woolley

Tinker, Printer, Solder, Die

These "printer wars" show how even newbies can get started into the fascinating and terrifying world of robotics with junk.

75 Then and Now

by Tom Carroll

ServoCity

There once was a time when robot builders had only a few sources for project parts. That is no longer the case.

Departments

06 Mind/Iron
Revenge of the
(Robot) Nerds

16 New Products

18 Events
Calendar

18 Showcase

66 SERVO
Webstore

81 Robo-Links

81 Advertiser's
Index

20 Bots in Brief

- In the Eye of the Hurricane
- Zoe Digs Life
- Walk Like a Man
- Fire See-er



34 PR2Lite Grows Up — and So Do Its Makers

by Alan Downing, Matthew Downing, and Frank Ou

Find out the latest evolution in this homebrewed version of Willow Garage's PR2.

40 Making Better Arduino Robots with the ArdBot — Part 1

by Gordon McComb

Originally debuted in 2010, ArdBot is back with a more slim line design, plus lots of added Arduinos for all kinds of new features and capabilities.

48 It's All in Your Head

by Morgan Berry

More and more, scientists are looking to the anatomy of the human body to make smarter and faster machines.



52 Tibbo Trouble

by Fred Eady

With Tibbo, you can write a few lines of Basic code and exchange data with peers and hosts via the Internet. The Tibbo EM500 is perfect for retrofitting devices with serial ports for use on the Internet. How can you implement it in your next bot build?

58 3D Printers Part 4: Tuning

by Michael Simpson

This time, slicer software that is used to convert 3D models to instructions a printer can actually understand will be covered, plus the MakerGear M2 arrived!



The Combat Zone...

24 BUILDER'S TIP:

You Can't Take It With You:
What Tools to Bring

27 RoboGames Requiem: A Facebook Epic

29 EVENT REPORT:

Robots Battle at the County Fair

32 PRODUCT REVIEW:

Snap Hubs

33 TIPS FROM THE PITS:

Getting Connected

by Bryan Bergeron, Editor

Revenge of the (Robot) Nerds

In the world of commercial robotics, brawn seems to have it over brains. The huge mechanical arms that bend, bolt, and bond metal don't have much in the way of computational capacity. Why supply a robot designed for dull, dirty, and often dangerous work with more than a smidgen of a brain? The first reason that comes to mind is safety.

There are useful reflex loops programmed into robotic TIG welders in some automobile assembly lines – including the ability to sense when contact has been made with a soft, easily damaged human. These safety mechanisms can be lifesaving.

Having worked with a desktop 3D printer for several months, I've come to appreciate the computational overhead of slicing and dicing a 3D model in such a way that it can be quickly rendered in 3D. Although this computation is largely performed on a desktop before porting the rendering files over to my 3D printer, the onboard processing is still magnitudes beyond the computational abilities of the CNC router that it replaced.

For robotics to really blossom, there has to be more computational intelligence available for navigation, sensing, human interaction, and self-repair. There's a good chance there will be more of this intelligence available in the next few years.

One reason for my optimism is the never-ending march of progress on the microcontroller front. Compare the Raspberry Pi to the popular Arduino, introduced only a few years ago. The price/performance point for the Pi simply blows away the Arduino – at least as far as raw computational ability goes. Then, there's the new Propeller chip and C compiler from Parallax which creates some interesting synergies with parallel processing hardware and a conventional programming language. There's probably a 1 GHz Arduino on the drawing board somewhere.

However, these are examples of evolutionary progress – the same sort of progress that we've seen in robotics for the past few decades. What about revolutionary, game-changing progress? When do we get even a glimpse of the robotic intelligence portrayed by Data in Star Trek or David in Prometheus?

There's a good chance we'll see it before the decade is out, thanks in part to projects such as the BRAIN (Brain Research through Advancing Innovative Neurotechnologies) initiative. One goal of this initiative is to develop new technologies that can record the activity of the brain at the level of individual neurons.

In contrast, much of our current understanding of the way the brain works is at a gross level, using tools such as functional magnetic resonance imaging (fMRI) that can map oxygen consumption by different regions of the brain.

Assuming the BRAIN initiative is funded, the AI and robotics communities should have much more data to work with. I expect the research findings to provide the basis for new algorithms and computational methods for intelligent robotics. Eventually, robots will become known for their computational prowess, and not simply as fancy R/C drones or replacements for semi-skilled human labor. **SV**

FOR THE
ROBOT
INNOVATOR

SERVO
MAGAZINE

Published Monthly By
T & L Publications, Inc.
430 Princland Ct., Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Webstore Only **1-800-783-4624**
www.servomagazine.com

Subscriptions
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277, N. Hollywood, CA 91615

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

**ASSOCIATE PUBLISHER/
VP OF SALES/MARKETING**
Robin Lemieux
display@servomagazine.com

EDITOR
Bryan Bergeron
techedit-servo@yahoo.com

CONTRIBUTING EDITORS

Jeff Eckert	Jenn Eckert
Tom Carroll	Kevin Berry
Dennis Clark	R. Steven Rainwater
Michael Simpson	Gordon McComb
Bryce Woolley	Evan Woolley
Fred Eady	Morgan Berry
Alan Downing	Matthew Downing
Frank Ou	Mike Jeffries
Andrea Suarez	Pete Smith

CIRCULATION DEPARTMENT
subscribe@servomagazine.com

**MARKETING COORDINATOR
WEBSTORE**
Brian Kirkpatrick
sales@servomagazine.com

WEB CONTENT
Michael Kaudze
website@servomagazine.com

ADMINISTRATIVE ASSISTANT
Debbie Stauffacher

PRODUCTION
Sean Lemieux

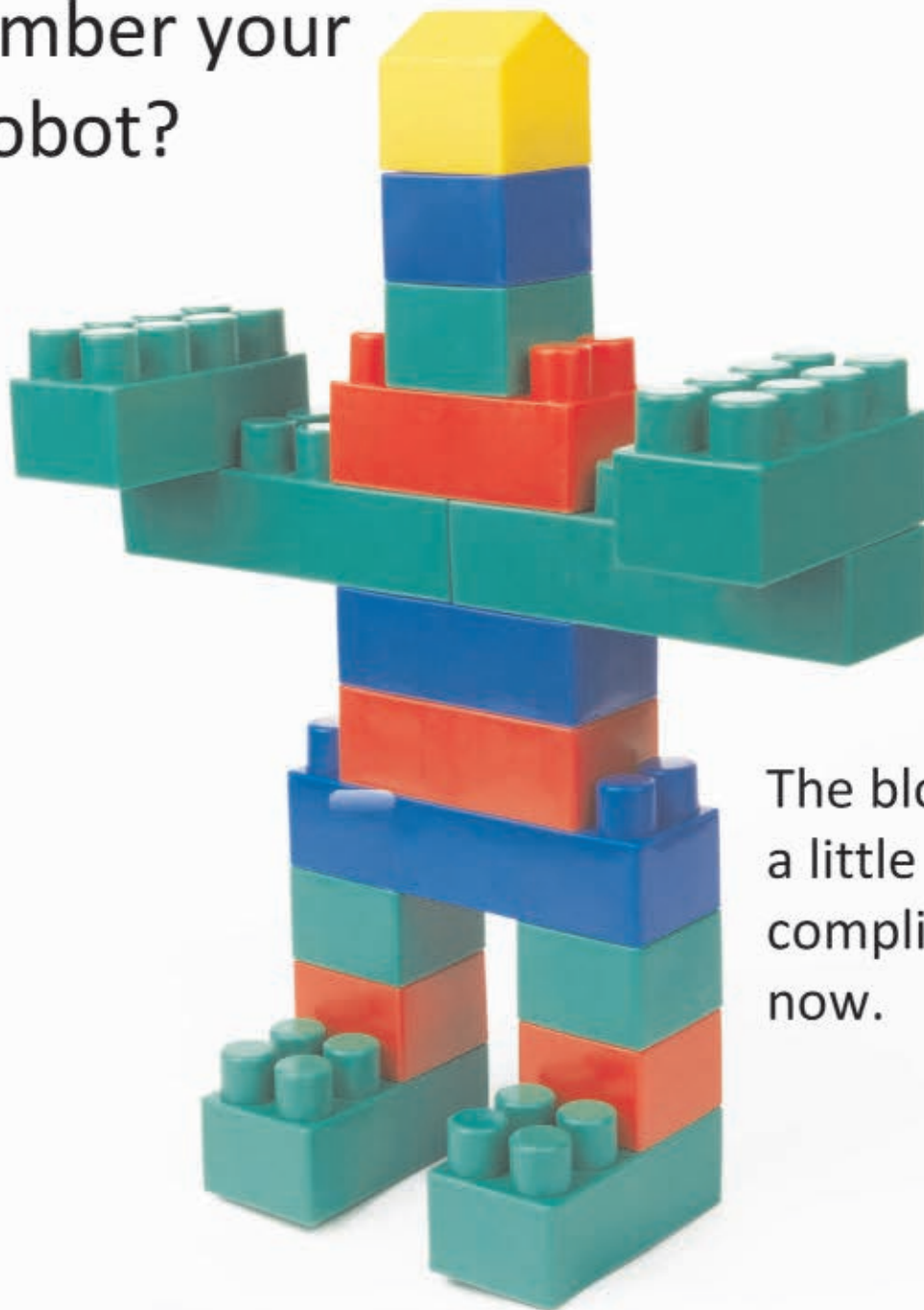
Copyright 2013 by
T & L Publications, Inc.
All Rights Reserved

All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *SERVO Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *SERVO*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *SERVO*. Please send all editorial correspondence, UPS, overnight mail, and artwork to: **430 Princland Court, Corona, CA 92879.**

Printed in the USA on SFI & FSC stock.



Remember your
first robot?



The blocks are
a little more
complicated
now.

for useful robot blocks visit us on the web, or find us on Facebook



hardware made easy

Solutions Cubed

designservices@solutions-cubed.com
www.solutions-cubed.com

open source hardware
Arduino compatible tilt
Compass wheel kit **NFC**
incremental encoder color sensor
load switch servo controls
object detector converters angle
motor controls



Robo Suit in Final Trials

Robotic exoskeletons designed to restore movement in disabled people have been around for several years, beginning with "Robot Suit HAL," introduced by Japan's Cyberdyne (www.cyberdyne.jp) back in 2008. As of last year, more than 300 HAL units were in operation, but only in Japan. However, it looks like a homegrown robo suit will be available in the USA by next year. Parker Hannifin — a major motion and control company — and Atlanta-based rehab clinic, Shepherd Center have formalized an agreement to commercialize Parker's Indego™ exoskeleton. Shepherd will take the lead in clinical testing, protocol development, and clinician training. Indego will not be widely available until sometime next year, but the first human patient has already been fitted with the device, allowing him to take his first steps in five years.

Few technical details have been offered, but it was revealed that the device weighs only 27 lb (11.3 kg) — about half the weight of competing units — and can be snapped apart to fit into a backpack. Its power output is adjustable, making it useful not only for people with complete spinal cord injuries but also for cases of stroke, MS, brain injury, and other conditions. According to Shepherd therapists, it can be used on all surfaces, including stairs. To see Indego in action, visit www.shepherdv.org/videos/39.

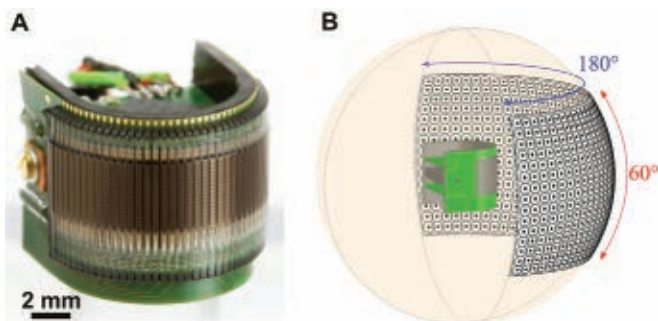


The Indego exoskeleton weighs only 27 lb and provides mobility on any kind of surface.

Insect Vision for Bots and More

Most machine vision systems are based on standard design cameras which, in turn, are based on the operation of the human eye. About 70 percent of all living species — including insects and even some mollusks — are equipped with compound eyes that operate quite differently. Typical resolution is pretty rotten — about 1/100th of our own resolving power. However, they are very good at detecting motion, and their large field of view is great for avoiding obstacles. The bottom line is that compound eyes are pretty much required for extremely small organisms (robotic or otherwise), as a human-type eye scaled down to fit a fly would not have a large enough aperture to admit a useful amount of light. Therein lies the concept of "CurvACE" (short for "curved artificial compound eyes") — a robovision system developed at the Swiss Federal Institute of Technology (EPFL, www.epfl.ch).

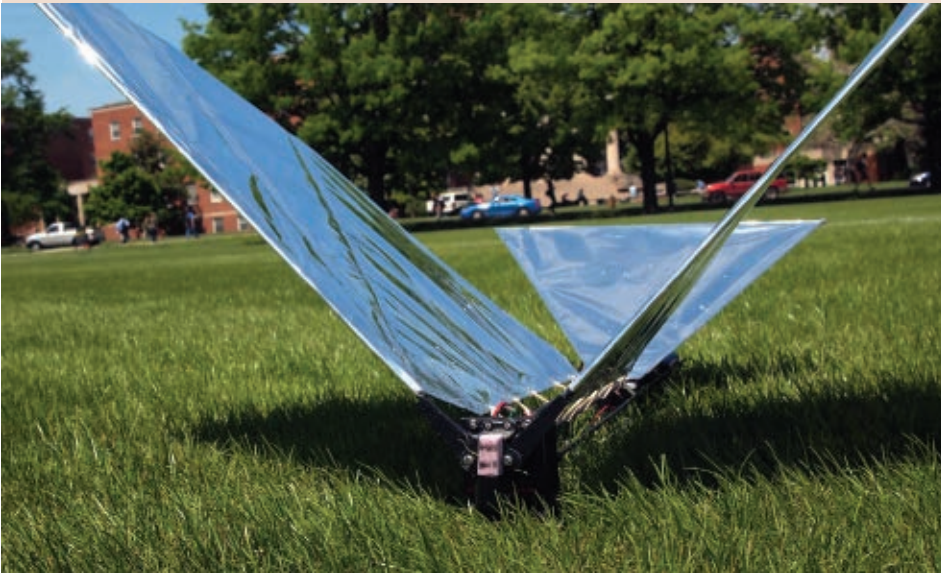
The system is constructed in three layers: a microscopic lens array; a light-sensitive array that resembles the circuitry of an insect brain; and a flex circuit for support and signal processing. According to EPFL, the device provides near-panoramic, undistorted views, picks up motion at 150 frames per second, and works in a range from bright sunlight to moonlight. The device occupies only 2.2 cm³, weighs 1.75 g, and consumes a maximum of 0.9W. Two of them — as you might expect — can be combined to provide a 360° view of the world. Projected applications include robots, driverless cars, and 3D imaging.



EPFL's prototype curved artificial compound eye.

Aerobic MAV Takes Flight

On the ground, it looks like a fourth grade origami project created with a roll of Reynolds Wrap™, but its creators at the University of Maryland's A. James Clark School of Engineering (www.eng.umd.edu) say that Robo Raven is a major breakthrough in micro air vehicle (MAV) technology. According to Profs. S. K. Gupta and Hugh Bruck, this is the first-ever wing-flapping vehicle capable of flapping its wings independently, which allows it to better mimic the aerobatic maneuvers of a real bird. The concept may seem both simple and obvious but, in fact, it requires the use of two programmable motors, an onboard microcontroller, and a bigger battery. The initial design turned out to be too heavy to fly, but designers cleared that hurdle by using "advanced manufacturing processes such as 3D printing and laser cutting to create lightweight polymer parts." They also developed improved motion profiles to give a better balance between lift and thrust.



"We can now program any desired motion patterns for the wings," Gupta noted. "This allows us to try new in-flight aerobatics — like diving and rolling — that would have not been possible before, and brings us a big step closer to faithfully reproducing the way real birds fly."

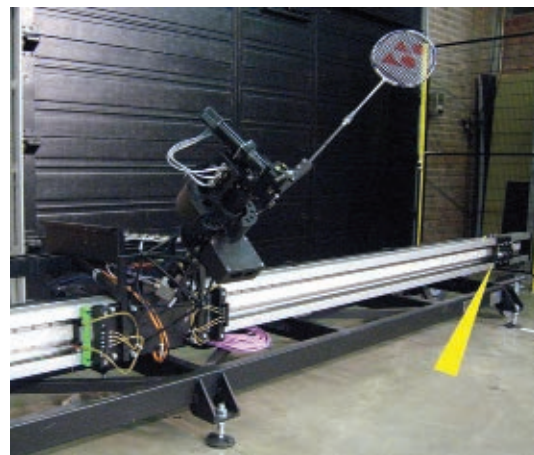
The result is pretty convincing, even to a local hawk that had a habit of attacking Robo Raven during trial flights. To see it (and the hawk) in action, just search "robo raven" on YouTube.

Robo Raven uses independent wing flapping to better imitate a real bird.

Bot Flips the Bird

Let's say you are a researcher working on the "Energy Software Tools for Sustainable Machine Design" (ESTOMAD) project at the Flanders' Mechatronics Technology Center (www.fmtc.be). Let's also say that you have received about €2 million to develop methods and tools to "model, simulate, analyze, and optimize energy flows and losses" in production machinery. Now you just need to come up with a suitable undertaking to accomplish and demonstrate that goal. What is the first thing that comes to mind? Why, badminton, of course. Hence, the world's first (and probably only) badminton robot, which appears in several vids on FMTC's YouTube channel (www.youtube.com/user/fmtcvzw).

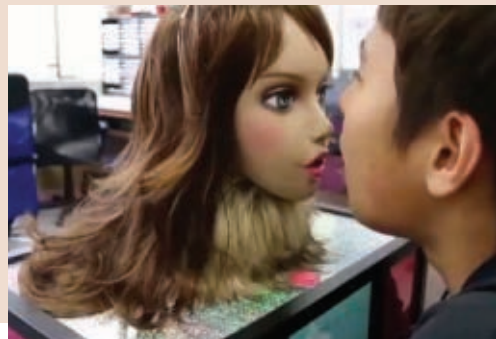
Well, nobody ever said that research can't be fun, and the ESTOMAD team reports that its design schemes are expected to produce an average lifetime energy savings of 30% in installed machines. Plus, the project members now have something to do on their coffee breaks.



The world's first badminton bot, created for Belgium's ESTOMAD project.

Sniff This!

Japanese roboticists have occasionally given us reason to wonder if they're totally insane. After all, some recent developments in the Land of the Rising Sun include a spankable robot butt and a coat designed to hug lonely people and whisper sweet nothings in their ears. Now, there is no longer any reason to wonder, as evidenced by recent projects from the "Crazy Lab" at the Kitakyushu National College of Technology (www.kct.ac.jp). One of them is a female robot named "Kaori," which also means "fragrance." Kaori's sole function is to detect foul breath. All you have to do is breathe in her face, and she will react with one of four possible comments, roughly translated as "No

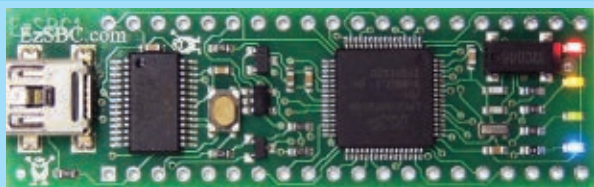


problem, Good sweet and sour," "It smells a little," "Intolerable, no good," and "It is over, this is an emergency situation."

Crazy Lab also came up with a dog version that crawls out of his doghouse to sniff people's feet. If your toe jam is particularly fetid, he pretends to faint. Rumors that Kaori soon will be an option on new Subarus cannot be verified. **SV**



Kaori the stink detecting bot, ready to sniff your breath.



Imagine It, Control It

with our 60MHz, 32-bit controller programmed in BASIC with full 64-bit floating point arithmetic.

32 IO Pins

I2C, SPI, UART

PULSOUT

Real Time Clock

PWM, SERVO

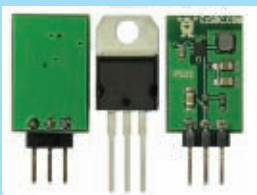
DHT11 Input

PULSIN

16x 10-Bit ADC

Buy one for \$49.95, get one free with code SERVO

3-Pin Switching Regulator
Delivers 5V or 3.3V at 1A
No Heatsink required!
Only \$6.95



EZSBC
Controlling your World the Easy Way

Visit us today @
www.EZSBC.com



AP CIRCUITS
PCB Fabrication Since 1984

As low as...

\$9.95
each!

Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards ship next day!

www.apcircuits.com



by Dennis Clark

Our resident expert on all things robotic is merely an email away.

roboto@servomagazine.com

You can discuss this topic at
<http://forum.servomagazine.com>.

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

I start this column knowing that Critter Crunch is only a short time away, and I need to get more testing for Silver Surfer going to make him more competitive. All the best intentions swallowed up by constantly being distracted ...

This month's question nicely coincides with my previous observation that walking robots are so "in" that I'm going to concentrate on them in the lab. There are two kinds of walkers that I'm attracted to myself: Humanoid RoboOne/Real Steel™ style robots and Mech Warrior™ style robots. Both are totally cool as far as I'm concerned. Lucky for us, they are both very buildable with today's technology!

I am a workshop hacker and don't have access to all of the knowledge and technology of the university projects, but I have Google and I'm not afraid to use it! So, while I am never going to get to the fancy coordinated flying quadcopters, I feel that a Mech walking around shooting the cat with LEGO dart cannons is not out of my league! How about you? What are your projects?

Regardless, robots are fun, cool, educational, instructional, and, let's face it, part of our society and mindset. They are here to stay! I drop into my kid's classrooms every year with robots and the children are totally into the discussion, especially the "hands-on" aspects when I let them play with the robots and get ideas. This year, a couple groups of kids made their own line follower mats to guide my little line follower robot. The bug bots are always a big hit and cause lots of squealing in the class when they go stomping around.

Now, on with the questions!

Q I have a RoboBuilder Creative kit that I've had for three years and have just gotten around to experimenting with it again. I've been working with the Motion Builder and Action Builder tools to make my own "moves," but the manual is not very clear about how to build custom motions. I am not having any luck at all with making an Action file. Can you help me Robiwan Kenobi? You may be my only hope!

— Trevor

A I've had a RoboBuilder 5720T (Figure 1) kit for a while now, and have built a few new

motions for it. You didn't specify which kit you had, so I'll assume that you have the Creator 5710K — the "basic black" one (Figure 2). (If you are interested in a good humanoid biped for a minimum of money, it is hard to beat this kit, which is typically under US\$450.)

In case folks aren't aware, the RoboBuilder Creator kit supports the construction of three basic types of walking robots: DINO, DOGY, and HUNO. Basically, these are dinosaur, dog, and humanoid. You can build other kinds of robots, of course, but then you break away from the graphically supported (in software) helper programs and the built-in

motion files that can't be changed.

You then need to go to the "custom" mode which makes everything a *lot* more difficult to do. (Getting a legged robot to walk is difficult!) The HUNO robot has 16 degrees of freedom (DOF) and moves pretty well for a biped with no lateral hip servos. Check out some videos on **www.robobuilder.net** for examples.

The manual is big on construction, but very, very thin on how to use its software. The **www.robobuilder.net** site has no software updates newer than 2009, but does have firmware updates up to 2.34.

Essential RoboBuilder Resources

Before I get started, let me point you to some essential resources if you are not conversant in Korean. The website **www.robosavvy.com** has the most comprehensive selection of manuals, software, firmware, and help that you will find anywhere on the RoboBuilder robot.

Here is the page where you need to start: http://robosavvy.com/site/index.php?option=com_openwiki&Itemid=&id=robobuilder.

This next page has discussions detailing the use of the firmware that the RoboSavvy folks have found and collected: <http://robosavvy.com/forum/viewtopic.php?t=3660>.

In fact, check out the RoboBuilder forums for lots of user information on utilizing the current firmware and replacing the whole software/firmware package with custom third-party code.

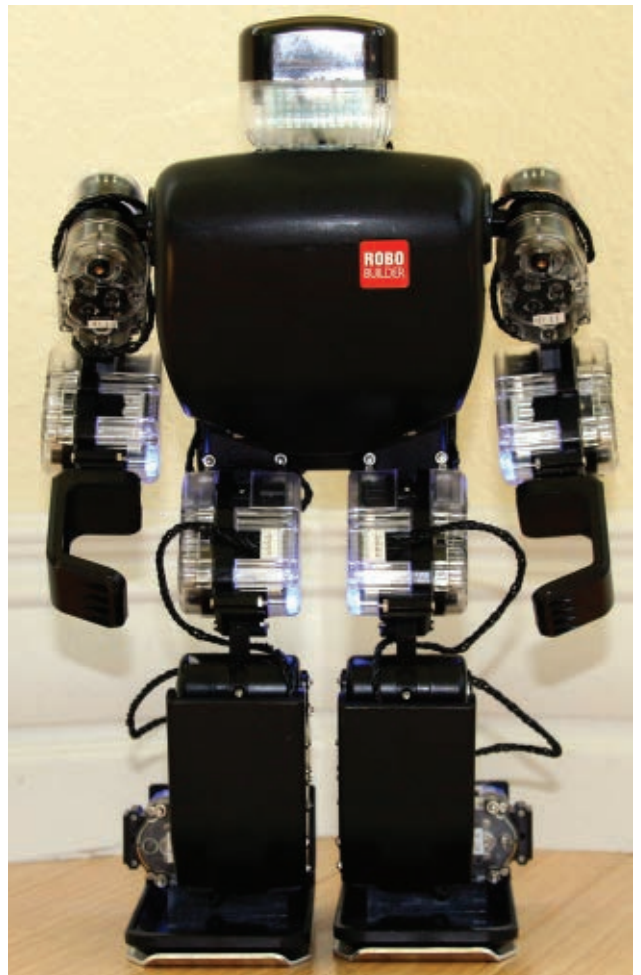
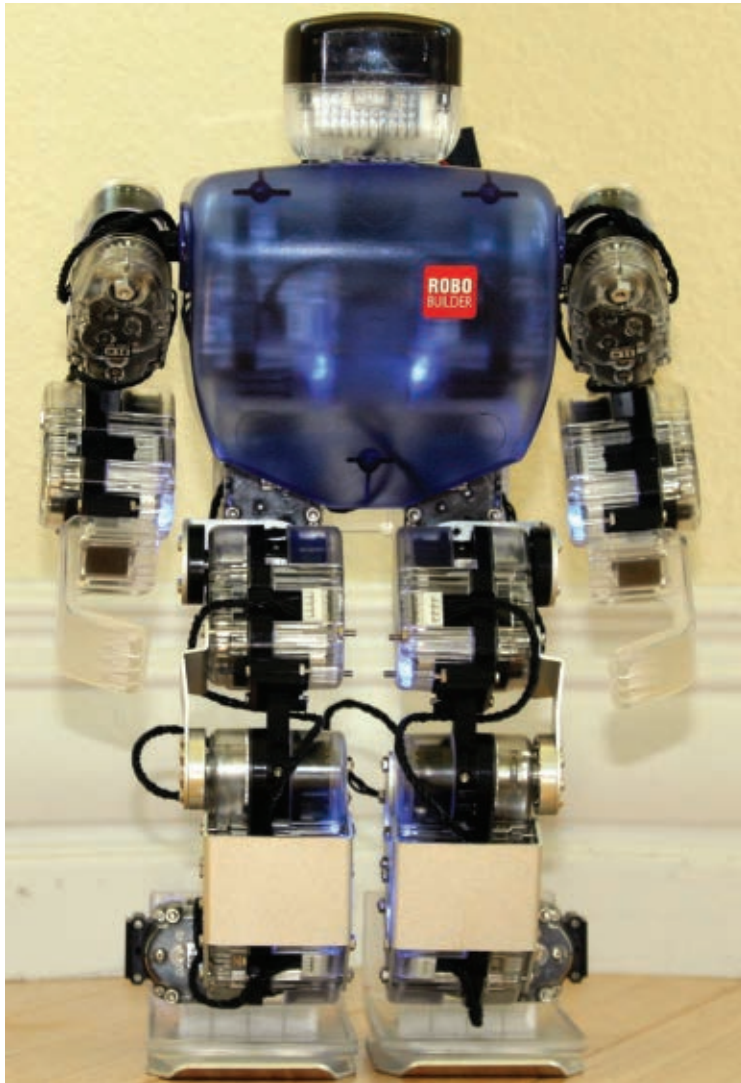


Figure 1

Figure 2

While I was researching this question, I queried the customer support folks at RoboBuilder and got a link to newer software on their "RQ" Korean-only site, which I shared with the RoboSavvy community on their Wiki site. This software is an upgrade to the software that comes with the robot.

Using Motion Builder

The first thing that you need to know is that all software is Windows only. I've run it on Windows XP Service Pack 3. Let's start with the demonstration motion file sent with the install.

Start up MotionBuilder from the "Start Menu" Programs selection.

You'll see **Figure 3** as your window. Click on the **Open** button and navigate to C:\Program Files\Robobuilder\Projects\HUNO\Motion Builder Files\HunoDemo.prj

Since the program now knows what platform you are using (HUNO), you will see a graphic of your robot to use as a reference; refer to **Figure 4**.

HunoDemo.prj is the project file in which your motion files are defined and collected together. It is essential that you define all of the motions you want to use together inside of a single project because when you download motion files in a session, any existing motion files in the RoboBuilder Controller (RBC) will be overwritten.

To open a motion file, click on the **Motion List** button at the upper right

of the program window. There are three buttons you are interested in (see **Figure 5**):

- **Add to Project** (Adds another motion file to the project.)
- **Open to Edit** (Opens the motion file for you to edit within the project.)
- **Delete from Project** (Kind of obvious ...)

Before we start editing, adding, or changing motion files, I've found that it is very useful to have your HUNO connected to both the computer and Motion Builder so you can get instant feedback. This step is also essential if you are going to "train" your robot with poses.

Select the **ComPort** you have connected your RoboBuilder serial cable to; the serial cable is the one with the DB9 connector on one end and what looks like a stereo earphone jack on the other. Then, click **OpenPort**. This button will turn green and then say **ClosePort** when communication has been established. If it doesn't work, check to make sure that none of the other RoboBuilder programs or any other program you are using has opened it already. Then, power-cycle your HUNO and try again.

You can now edit and enhance this motion file. Chapter 3 of the *RoboBuilder User's Guide* gives the basic details of the use of the program's edit and save buttons. When it comes to making totally new motions, you'll want to start with your "neutral" setting that is configured into your robot's initial setup.

My next steps are to think through the complete sequence of actions like I was doing an old-fashioned stop-motion animation; then start capturing poses that represent the end point of each "scene" (see **Figure 4** for the **Get Pose** button).

When you press the **Get Pose** button, you'll get the window shown in **Figure 6** which allows you to select which servos you will be posing. Check the HUNO graphic with all of the servos numbered and located to find the ones you want.

When you are getting your initial pose, if it is a "whole body" change, you'll want to select all of the servos. You can tweak your pose later by selecting only those servos you want to change, or you can use the "dials" next to the servo you want to tweak.

If you have your HUNO connected, tweaking these dials will have an immediate effect on your robot. These functions make the creation of new motion sequences really simple and kind of fun.

While you are capturing a pose, the affected servos have no power so you'll have to be careful and hold it all together; when ready, press the **Capture** button. This is what the

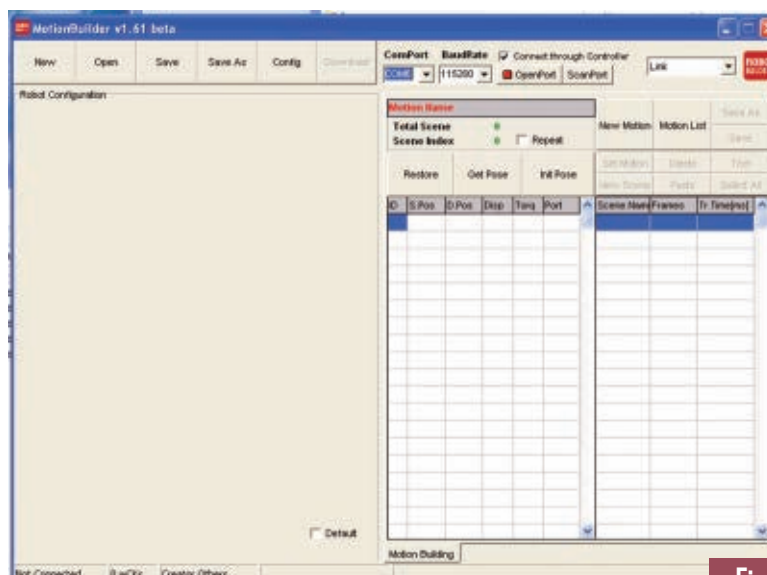


Figure 3

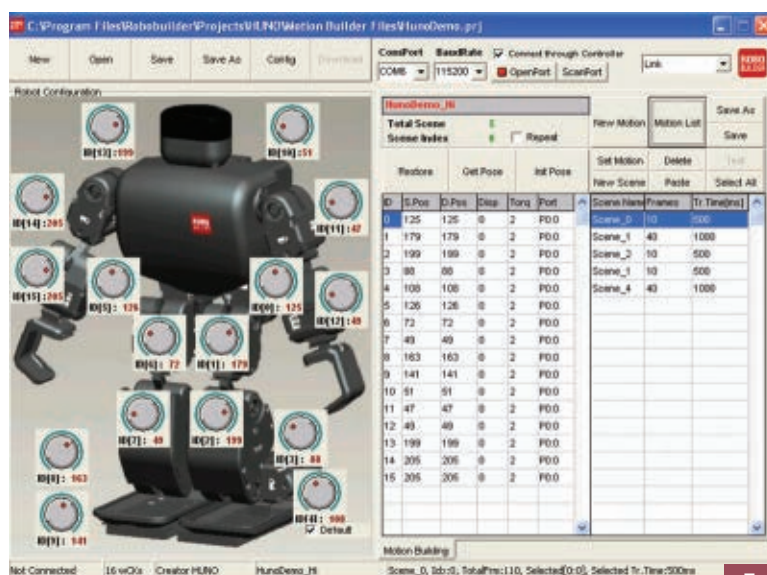


Figure 4

Get Pose button turns into while in Capture mode. When doing a complex capture, you'll find you need a third hand to push the buttons; you can do it by yourself if you are careful and move slowly.

Each scene will start off where the previous scene ended. To transition between one scene and the next, you have to tell Motion Builder how many frames to use and how long you want the transition to take. The wCK servos do not have speed settings, so when you are moving

between scenes Motion Builder will divide the time and distance for each servo up by the number of frames that you want to use.

Each frame must take at least 20 ms; if you try to put more frames in than will fit, Motion Builder will restrict your frame rate or your time to make it work out. Experiment with this and you'll quickly learn what frame count and times give you the fastest and/or the smoothest transition rate.

You can tweak your end positions

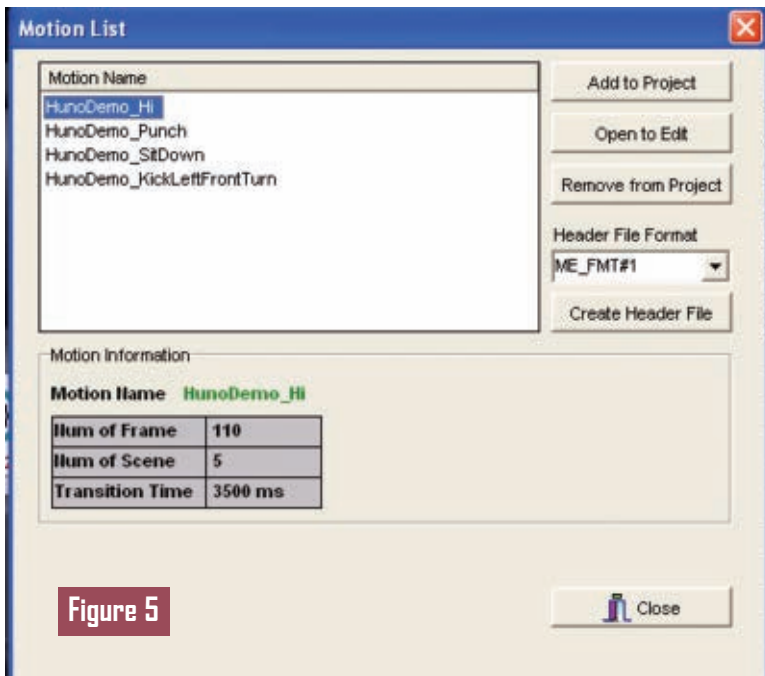


Figure 5

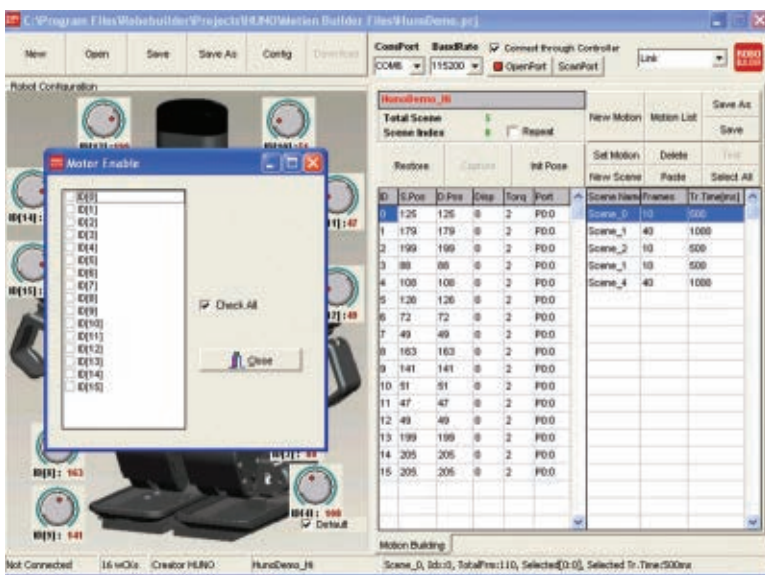


Figure 6

by changing the **D Pos** column values until you get what you want. You can also use the dials next to the servos you want to change if you like a more graphical feedback process. If you ever goof up too badly and forget where you started, just click the **Restore** button and all the changes since your last save will be returned.

The **Init Pose** button allows you to set the initial pose for all of your

motions – I have not messed with this, but eventually, I’m sure I’ll find a use for it.

Sending Motion Files to the RBC

Eventually, you will be done playing with your motions and will want to un-tether from your computer and go show off. The latest software

includes a nifty program called RoboBuilder Tool which calls itself RoboBuilder Download Tool when it’s launched. This tool will allow you to collect motion files from any project and arrange them in the proper order for your remote control buttons.

The first file goes to button 1, and so on. If you have more than 10 motions, the next 10 motions are assigned starting with *1, and so on. I like to arrange things so that “left side” motions go on numbers 1, 4, and 7; “right side” motions go on 3, 6, and 9, and the center column numbers get forward and back motions. **Figure 7** is a screenshot of the Download application; notice the limit of 20 motion files.

Make sure that you have closed your COM port on Motion Builder. Now, you can open the COM port on RoboBuilder Download Tool. Using the **Directory** pane, navigate to the folder with your motion files. You will see your files appear in the **File** pane. You can select each motion and move it to the **Download File List** with the single arrow (>) so that you can put your motion files in the order you want. Or, you can just use the >> to move all the motion files over, and then use the **Up** and **Down** buttons to re-arrange them once they are added. You can add the same motion file over multiple times if you wish.

Once you have all of your motion files in the order that you want them, press the **Download** button. It only takes a second or two to download all the motion files. You can test your order by using the **Remocon** pane which simulates the buttons on your IR remote.

This should keep you busy for a while. I am experimenting with other programs and options in the Motion Builder and Action Builder programs for RoboBuilder. There are some glaring omissions in Motion Builder – like the ability to mirror a motion or a scene completely (more about that next month) – and Action Builder promises more than it delivers, unfortunately, as there are some actions that I consider essential that

Next month, I'll write more about Motion file nuances and how to define Action files for the RoboBuilder.



Send your questions to roboto@servomagazine.com and I'll do my best to answer them! **SV**

SERVO 08.2013 **15**

NEW PRODUCTS

Aluminum End Caps

ServoCity's new aluminum end cap B (#545448) fits into the end of their 1" x 1.5" tube gearboxes and allows users to attach any of their 0.770" pattern components. The end cap has a 1/2" center bore to allow tubing or shafting to extend into the inside of the gearbox for added support. When using hollow tubing, it allows wires to be routed through the interior of the gearbox and down the hollow tube for a clean look. When used to join a tube gearbox to a bottom mount gearbox, it creates a setup for an antenna tracking system. This end cap is constructed of 6061-T6 aluminum for superior strength; price is \$4.99/each.



1/4"-20 Hub Mount

The new 90° quad hub mount E (#545452) from ServoCity enables a 1/4-20 bolt and nut to be used with their aluminum channel and other Actobotics™ components. Four holes are counter-bored for 6-32 socket head cap screws, while the remaining four holes are 6-32 tapped. This hub mount makes it easy to attach a camera to a robotic platform, drone, or other structure. It is machined from 6061-T6 aluminum for superior strength and durability. The 1/4-20 nut is not included; price is \$5.99/each.

For further information, please contact:

Sonic Shifter

AndyMark, Inc., announces the Sonic Shifter — their newest shifting gearbox. This shifting, two-speed transmission offers eight different gearing ratio options to suit applications.

AndyMark has provided two speed transmissions for mobile robots since 2004. The AM Shifter and Super Shifter have been used by many *FIRST* Robotics Competition teams. Each of these transmissions has matured through multiple design iterations. AndyMark has taken what they learned from these two dependable transmissions to make the Sonic Shifter.

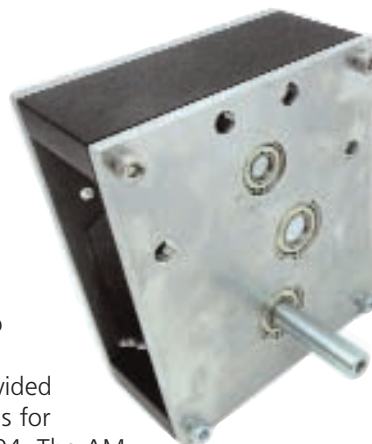
The Sonic Shifter is a shift-on-the-fly gearbox which uses many of the same features as the AM and Super Shifters. The same durable dog gear shifting method is used, along with the same motor input and shaft output designs. This transmission can still be shifted by using pneumatic or servo shifting mechanisms. The improved features of the Sonic Shifter include:

1. The output shaft is centered on the shaft plate, allowing a designer to make replica parts from the left to right side of their robot drive train.
2. The housing for the Sonic Shifter now has openings on each side, allowing for easy debug, maintenance, and lubrication.
3. Additional gear ratios are available through the Sonic Shifter, enabling designers and operators more choices in their robot's speed and torque.
4. Lightening features have been implemented, allowing for weight savings compared to other AndyMark shifters.
5. A more robust servo shifting method is available compared to prior servo shifting methods. This servo shifter pushes with 8.5 pounds of force compared to the AM and Super Shifter's force of 5.5 pounds.

Many mobile robots depend on the need of a shifting transmission. There are three reasons these transmissions are used in competition robotics:

1. High speed and high torque.

Selection of high speed or high torque can be optimally attained while only utilizing a limited amount of motor



power. Often, a robot driver (or autonomous program) switches to high speed in order to make a quick dash to get to an important location on the playing field before a competitor, for example. Also, it is good to have a high torque selection available in order to push through obstacles.

2. Better position control.

During critical parts of a robot competition, low gear is needed for quick acceleration or for fine-tuned robot placement. If the robot was only geared for a fast speed, incremental robot positioning is more difficult and the robot controller has less ability to place the robot in a required location.

3. Better power management.

Robot competitions are limited by time, amperage draw, and battery life. Near the end of a match or during a series of multiple matches, power management systems for robots become overloaded. A robot with more motor input power geared at only a high speed consistently draws more current than a robot using less motor power but utilizing a shifting transmission. For robots using a shifting transmission, robot controllers often shift their transmissions to low gear when a robot's circuit breakers are beginning to trip under a high power load. Shifting to a lower gear which provides a mechanical advantage for the drive train) will draw less current, saving the battery life and enabling the robot to continue to operate.

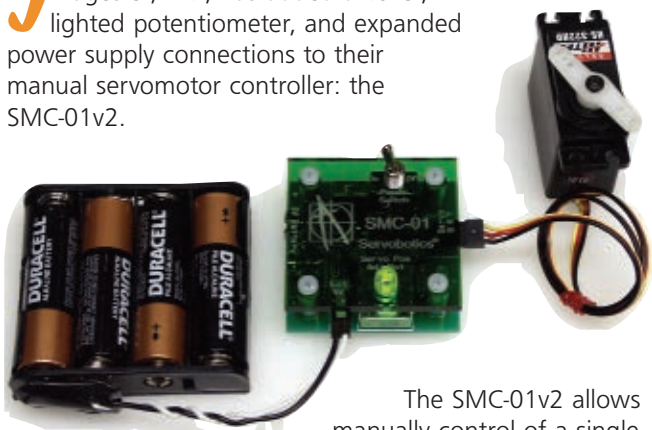
For further information, please contact:

AndyMark Inc.

Website: www.andymark.com

Manual ServoMotor Controller

Images SI, Inc., has added a cover, lighted potentiometer, and expanded power supply connections to their manual servomotor controller: the SMC-01v2.



The SMC-01v2 allows manually control of a single servo motor. The knob

proportionally controls the servo motor position. The servo response is in proportion to the speed and position of the potentiometer knob.

Green LED lights the potentiometer knob. A right-angle three-pin header connects servo motors. The circuit is controlled by a PIC microcontroller. The SMC-01v2 may be

powered by a nine volt battery or a 5-6 volt rechargeable battery pack. It can be purchased as a kit or assembled.

List price is \$39.95 for the assembled unit; the list price is \$29.95 for the kit (soldering required).

For further information, please contact:

Images, Co.

Website: www.imagesco.com

Hypermode Series of Controllers for N64, NES, SNES, and Genesis

Innex, Inc., has launched a full line of wireless controllers by Retro-Bit® that are designed specifically for old-school video game platforms. Not only is each controller wireless, but is also engineered to allow for turbo-type functionality to be used with their respective platforms. A couple common features include the ability to program each button to normal, auto fire, or turbo mode, and trigger buttons.

Other product highlights include:

Wireless Hypermode Series Controller for N64 (\$29.99)

- Compatible with N64 consoles.
- Classic analog stick, D-pad, and button layout.
- 2.4 GHz wireless controller with receiver.



- Open slot for Rumble or Memory Pak.
- Requires three AAA batteries.

Wireless Hypermode Series Controller for NES/SNES/PC-MAC (\$29.99)

- Compatible with NES, SNES, PC, and MAC.
- Similar to the Famicom color scheme.
- Three connecting ports SNES, NES, and PC/MAC via USB.
- 2.4 GHz wireless controller with receiver.
- Built-in rechargeable battery.



Wireless Hypermode Series Controller for GENESIS/PC-MAC (\$29.99)

- Compatible with GENESIS, PC, and MAC.
- Similar to the six-button Genesis controller scheme.
- Two connecting ports for GENESIS and PC/MAC via USB.
 - 2.4 GHz wireless controller with receiver.
 - Built-in rechargeable battery.



For further information, please contact:

Innex, Inc.

Website: www.innexinc.com

EVENTS

ROBOTS.NET

Know of any robot competitions I've missed? Is your local school or robot group planning a contest? Send an email to steve@ncc.com and tell me about it. Be sure to include the date and location of your contest. If you have a website with contest info, send along the URL as well, so we can tell everyone else about it.

For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: <http://robots.net/rcfaq.html>.
— R. Steven Rainwater

AUGUST

5-8 **AUVS International Aerial Robotics Competition**
Betty Engelstad Sioux Center

UND, Grand Forks, ND
University teams field fully autonomous air vehicles and sub-vehicles that must perform a variety of tasks.
[http://iarc.angel-strike.com](http://iarc angel-strike.com)

8-18 **Missouri State Fair Robot Expo**

*Missouri State Fair
Sedalia, MO*
4-H, state universities, and other organizations will be holding a variety of robot contests, demonstrations, and hands-on tutorials throughout the fair. Check the fair schedule for times and locations.
www.mostatefair.com

11-13 **CIG Car Racing Competition**

Niagara Falls, Ontario, Canada
Every year the AAAI participants design autonomous controllers for robot racing cars.
<http://games.ws.dei.polimi.it/competitions/scr/>

24-29 **FIRA Robot World Cup**
Bristol, England
Every kind of autonomous robot soccer you could want.
www.fira.net

30 **DragonCon Robot Battles**
Atlanta, GA
Autonomous and remote-controlled robots battle at the famous Georgia science fiction convention.
www.dragoncon.org

Recycling & Remarketing High Technology
WEIRDSTUFF® WAREHOUSE
Software, Computers, Electronics, Equipment, Doo-hickies
**384 W. Caribbean Dr.
Sunnyvale, CA 94089**
Mon-Sat: 9:30-6:00 Sun: 11:00-5:00
(408)743-5650 Store x324

 **WE BUY AND SELL EXCESS & OBSOLETE INVENTORIES!**

FREE COMPUTER RECYCLING
We recycle computers, monitors, and electronic equipment. M-Sat 9:30-4:00 

 **GREAT DEALS!**
Hi-tech items, electronics test equipment, and more!

GIANT AS-IS SECTION
10,000 sq. ft. of computers, electronics, software, doo-hickies, cables, and more! 

also check out our...
eBay Store
stores.ebay.com/WeirdStuff-Inc
WWW.WEIRDSTUFF.COM

ALL ELECTRONICS CORPORATION

Thousands of Electronic Parts Available Today!

LEDs · CONNECTORS · RELAYS
SOLENOIDS · FANS · ENCLOSURES
MOTORS · WHEELS · MAGNETS
PC BOARDS · POWER SUPPLIES
SWITCHES · LIGHTS · BATTERIES
and many more items...

We have what you need for your next project.

www.allelectronics.com
Discount Prices · Fast Shipping

SERVO

SERVO BOOK SPECIALS

Raspberry two book combo

 **Programming the Raspberry Pi**
Getting Started with Python
Simon Monk

 **Raspberry Pi User Guide**
Eben Upton
Raspberry Pi Foundation

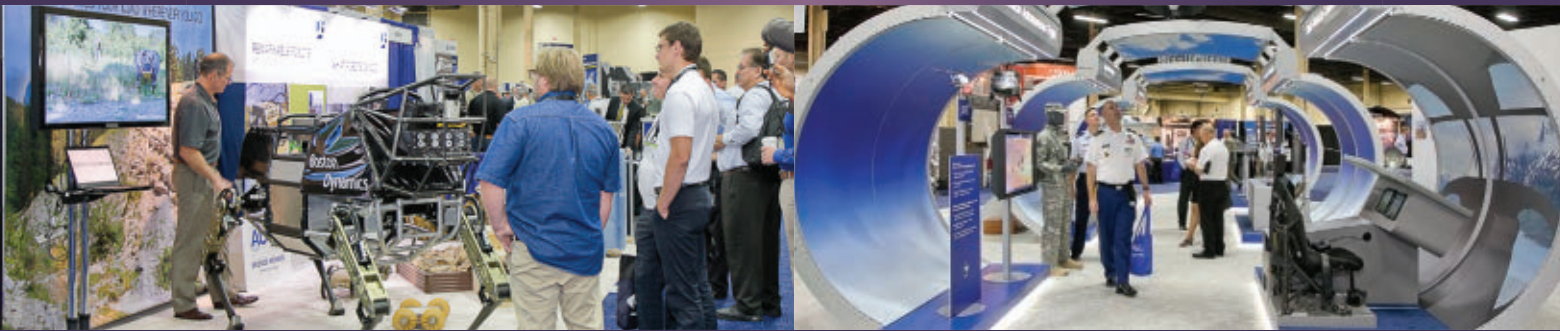
Free STD Shipping!

 **Getting Started with chipKIT™**
The Arduino Compatible PIC32 based module
Chuck Hollnagel

For more specials and subscriber discounts please visit:
<http://store.servomagazine.com>
Offer Ends 8/31/13



WASHINGTON, D.C.
WALTER E. WASHINGTON CONVENTION CENTER
CONFERENCE: 12-15 AUGUST
TRADE SHOW: 13-15 AUGUST



With more than 40 product categories and 600 exhibitors, you will find no shortage of innovation, technology and products in the unmanned systems and robotics industry.

Imagine being in one place with the biggest influencers in the industry who all want to share their most valuable knowledge with YOU!

- **Learn** about important issues, advancements and opportunities in the unmanned systems and robotics industry.
- **Choose** from more than 100 technical sessions, panels and workshops featuring the industry's top speakers and leading experts.
- **Explore** the latest technology on the market, including live air, ground and maritime demonstrations, and much more.
- **Mingle** and network at daily receptions, coffee breaks, generous exhibit hall hours and our premier networking event, The Mix.

WILL YOU BE THERE?

Register today at **auvsishow.org**

bots IN BRIEF

IN THE EYE OF THE HURRICANE

Hurricanes are generally things that robots, humans, and everything else try to avoid. So, it's kind of hard to study them. There are some aircraft that are specifically designed for hurricane study but they're big and expensive, and since they're stuffed full of humans, they can't do anything particularly risky. Such dangerous tasks are best left to robots, like this chubby little guy from the University of Florida.

These robots are made of carbon fiber, are just six inches long, and don't weigh much more than an iPod Nano. Hardware on board measures pressure, temperature, humidity, location, and time — all to help predict the trajectory and intensity of hurricanes. Since just one of these robots isn't nearly enough to get a good sense of a hurricane, they're designed to be used in swarms of tens or hundreds, collecting massive amounts of data while creating their own autonomous network.

The most interesting part of this system is how the robots get around. "Our vehicles don't fight the hurricane; we use the hurricane to take us places," said [Kamran] Mohseni, the W.P. Bushnell Endowed Professor in the department of mechanical and aerospace engineering, and the department of electrical and computer engineering.

The aerial and underwater vehicles can be launched with commands from a laptop hundreds of miles from the eye of a hurricane. Mohseni and a team of graduate students use



mathematical models to predict regions in the atmosphere and ocean that can give the vehicles a free ride toward their destination. Once in the vicinity, they can be powered off to wait for a particular current of wind or water. When they detect the current they need for navigation, they power back on, slip into the current, then power off again to conserve fuel as the current carries them to a target location. In essence, they can go for a fact-gathering ride on hurricane winds and waters.

At just \$250 each (less if they're produced in bulk), the robots are inexpensive enough that you can plan to lose a few (or a whole bunch) since the data they collect are (hopefully) worth far more than the cost of the robots.

Smart systems of robust and borderline disposable robots are good for a lot more than climate monitoring, too. Think about how valuable such systems would be for mapping or search and rescue applications. The trick has always been finding the ideal mix of autonomy, capability, and low cost, but now that folks are able to build robots that cost just a few hundred bucks, it makes sense to start using them (and losing them) in place of manned systems.

bots IN BRIEF

ZOE DIGS LIFE

Finding life on planets is a tricky business, as evidenced by the fact that we've so far completely struck out everywhere except our own backyard. It's going to take some practice to figure out where and how to look, which is why a robot named Zoë is heading back to the Atacama Desert in Chile.

The Atacama Desert is a brutal place. It's mostly waterless and so high up in elevation that the air is thin and solar radiation is significantly higher than normal. Not much lives there. About the only things that can survive are micro-organisms, and even they have to hide beneath the surface. This is about as close as you can get to a planet like Mars, and it's Zoë's job to test out instruments and techniques that could lead to the discovery of life on the red planet.

Field investigation over three years will use a rover to make transects of the Atacama with instruments to detect subsurface micro-organisms and chlorophyll-based life forms, and to characterize habitats. The rover will integrate panoramic imagers, microscopic imagers, spectrometers, as well as mechanisms for subsurface access. Robotic considerations in addition to instrument integration include platform configuration, planetary-relevant localization, complex obstacle negotiation, over-the-horizon navigation, and power-cognizant activity planning. An architecture that coordinates these capabilities, provides health monitoring and fault recovery, and allows for variability in the degree of



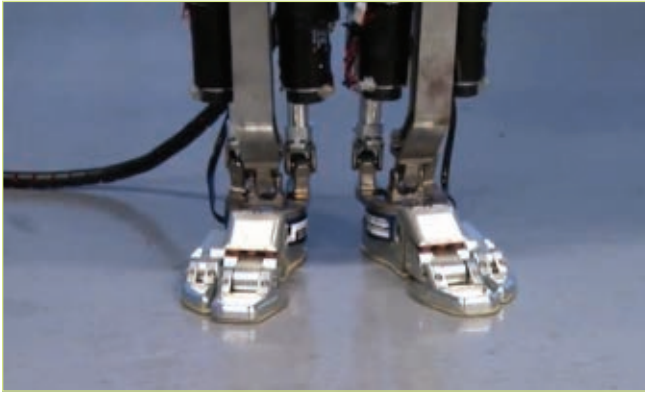
autonomy is vital to long-duration operations.

The measurement and exploration technique produced by this investigation combines long traverses, sampling measurements on a regional scale, and detailed measurements of individual targets. When compared to the state-of-the-art in robotic planetary exploration, this approach will result in a dramatic increase in the number of measurements made and data collected by rover instruments per command cycle. This result will translate into substantial productivity increases for future planetary exploration missions.

Previous studies focused on finding life on the surface, but especially on Mars, the subsurface is where it's at. Zoë, created by a team led by Dr. David Wettergreen from Carnegie Mellon University's Robotics Institute, might not look as fancy as the Curiosity rover, but it's packing a bunch of science under its solar panel hood, and this time around, there's a fancy new piece of hardware. It's got a drill.

The drill comes from Honeybee Robotics, and samples that it retrieves will be analyzed by the Mars Microbeam Raman Spectrometer — an early candidate for the 2020 Mars mission (this one). Zoë started her mission just recently, and should take one or two drill samples every day. She's got 50 kilometers to cover autonomously in two weeks, and you can keep up with her progress at <http://frc.ri.cmu.edu/atacama/index.php>.





WALK LIKE A MAN

Since bipedal robots took their first steps, the majority have been designed with the same basic joint/actuator configuration in their legs. This design — based on a simplified human leg — uses just six motors (three for the hip, one for the knee, and two for the ankle), and though it proved successful, it has also shown several limitations over the last 25 years. Now, researchers at the Humanoid Robotics Institute at Waseda University (the birthplace of the first real humanoid robots) have set out to reinvent the wheel (actually, the leg) by developing an entirely new shank that more closely replicates human walking.

The researchers — led by Professor Atsuo Takanishi — presented their groundbreaking work at the IEEE International Conference on Robotics and Automation (ICRA) earlier this year in Germany.

To design and test their new calf "muscles," the group started by taking a closer look at the legs of one of their robots — WABIAN-2R (WAseda Bipedal humANoid - No. 2 Refined) — which has been regarded as one of the world's most sophisticated humanoid robots since it was unveiled in 2006. It stands 148 cm (four feet 10 inches) tall, weighs 64 kg (141 pounds), and — with its 41 degrees of freedom — can perform very human-like movements. What sets it apart from many other bipeds is its flexible pelvis which gives it the ability to walk with stretched knees (unlike Honda's ASIMO, among others).

Most biped robots walk with flat feet that land parallel to the ground, but WABIAN-2R's feet — with curving arch and flexible toes — lands heel-first and lifts off at its toes. That's progress, but a few vexing inconsistencies remain: Humans walk with their feet roughly 90 mm (3.5 inches) apart — a distance doubled by the robot due to its large ankle motors. When walking, the robot's center of mass had a lateral movement of 50 mm (compared to just 30 mm for a human). It can't mimic a person's foot rotation (approximately 12 degrees) due to a missing yaw joint in its ankle; a problem it and other biped robots inherited through the old joint configuration.

So, with those limitations in sight, the Waseda

researchers started working on their new leg. The missing foot rotation could come from the yaw axis in the hip, but that would also rotate the knee and just introduce its own set of problems. Instead, the researchers decided they would have to add a yaw joint somewhere below the knee, while at the same time shrinking the lower leg's overall size to shorten the step width between the feet. They began by looking at the average size and movement range found in humans, and analyzed motion-capture data to determine the optimal performance requirements of their new ankle.

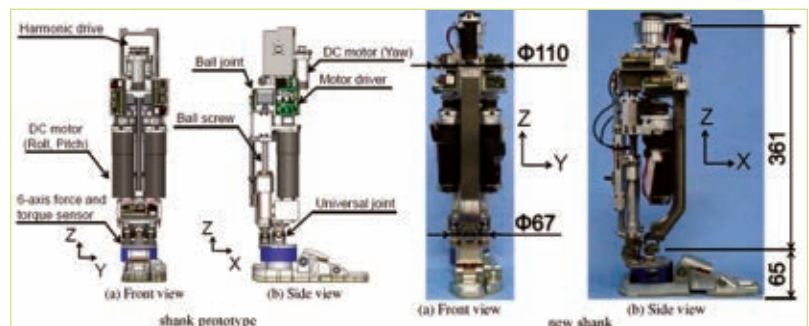
In designing the prototype, the yaw rotation would be provided by a motor located at the top of the shin, just under the knee. The two motors normally situated in the ankle for pitch and roll were ditched in favor of a linear parallel link mechanism which trimmed precious millimeters off the lower leg's circumference. Here is where they ran into their first major problem, however. There was no commercially available ball joint strong enough to withstand the compressive loads it would need to support. Their solution was to design one from scratch with the help of Japanese company, Hephaist Seiko. (The project was supported in part by the Humanoid Robotics Institute, RoboSoM project, and Japan's MEXT.)

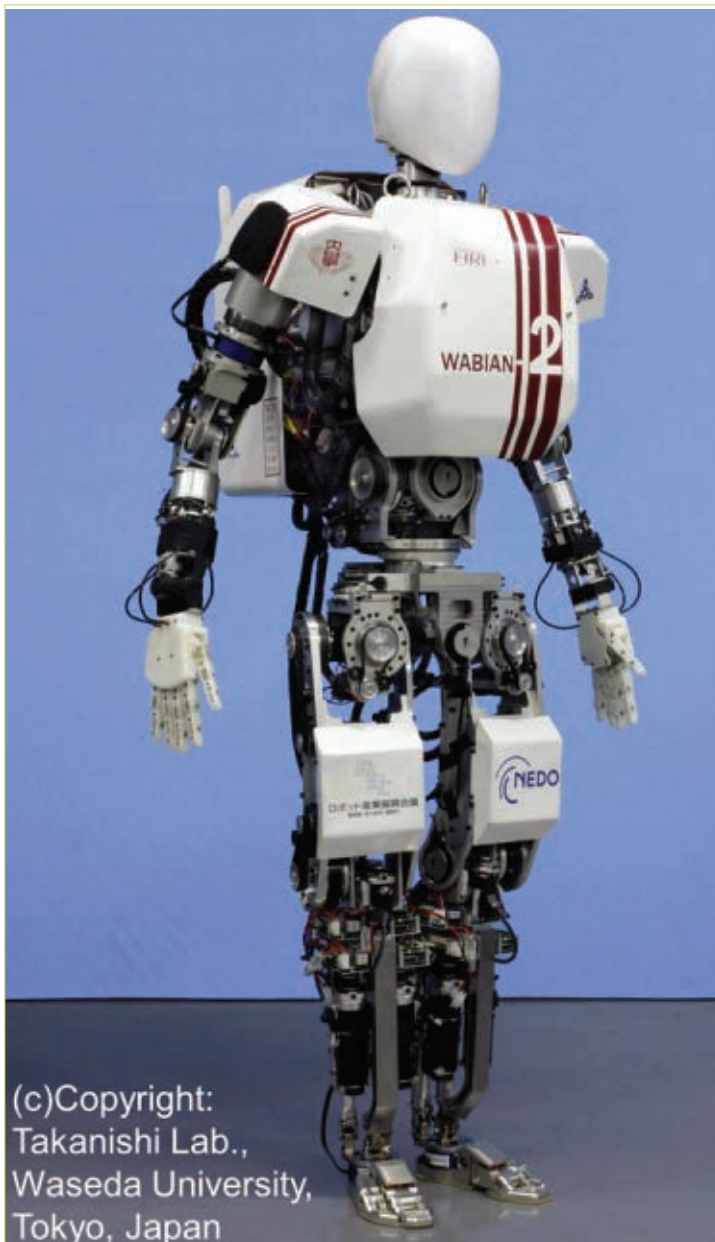
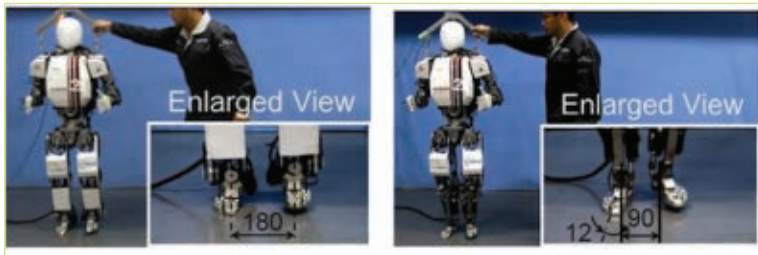
The shank prototype's universal joints were causing problems and needed to be redesigned. Minor tweaks to the universal joints and material changes resulted in vastly improved stiffness and increased movable range. These and other refinements led to an entirely new system. The researchers installed the new shank in their humanoid robot, which is now called WABIAN-2RIII.

This time, the robot worked marvelously. In experiments where it walked in place, not only was it able to match the human step time of 0.6 seconds, it also matched the required 12 degree foot rotation and 90 mm step width. The researchers have also successfully reduced its center of mass lateral movement from 50 mm down to 34 mm (just 4 mm shy of an actual human).

Next, the team plans on experimenting with forward walking, and figure they can make adjustments to their existing walking pattern to achieve more realistic human-like walking.

The researchers are: T. Otani, A. Iizuka, D. Takamoto, H. Motohashi, T. Kishi, P. Kryczka, N. Endo, L. Jamone, K. Hashimoto, T. Takashima, H.O. Lim, and A. Takanishi.





FIRE SEE-ER

This firefighting robot from UCSD (University of California, San Diego) is doing something that firefighters would find immediately useful.

The UCSD robot is called FFR for "firefighting robot," although FLR for "fire locating robot" might be more technically correct. The robot uses both a stereo and a thermal camera to generate 3D PointClouds with thermal overlays, allowing the robot to autonomously generate maps showing hot spots and humans even through smoke. The sensor hardware on board the robots doesn't look especially complex, meaning that the 'bots might ultimately become inexpensive (and replaceable) enough to deploy in swarms. So, instead of running around burning buildings looking for people, firefighters can deploy a bunch of robots first, and rapidly build up a thermal map telling them where to go.

*Cool tidbits herein provided by
www.botjunkie.com, www.plasticpals.com,
<http://www.robots-dreams.com>, and other places.*

COMBAT ZONE

Featured This Month:

- 24** **BUILDER'S TIPS:**
You Can't Take It With You: What Tools To Bring
by Mike Jeffries
- 27** *RoboGames Requiem: A Facebook Epic*
by the Robot Combat Community
as told by Kevin "Legendary Robotics" Berry
- 29** **EVENT REPORT:**
Robots Battle at the County Fair
by Andrea Suarez
- 32** **PRODUCT REVIEW:**
Snap Hubs
by Pete Smith
- 32** **CARTOON**
- 33** *Tips from the Pits: Getting Connected*
by Pete Smith

Go to www.servomagazine.com/index.php?/magazine/article/august2013_Combat_Zone for any additional files and/or downloads associated with this article. You can also discuss this topic at <http://forum.servomagazine.com>.

BUILDER'S TIPS

You Can't Take It With You: What Tools To Bring

● by Mike Jeffries

Your robot is (hopefully) done, and it's time to pack your bags and make your way to the event. While it would be handy, there are quite a few impracticalities to bringing all of your tools and equipment with you. This means you'll need to bring a selection that at least should be enough to get you through the event without spending too much time scrambling for tools and borrowing other team's gear.

I won't be talking about tools that are particularly specialized. If your machine needs any specific equipment for basic maintenance and repairs, those take priority. The following sections will cover tools that are almost universally useful in the pits during an event.

The Bare Essentials

This section covers the smaller equipment that should be easily crammed into a relatively small toolbox.

Hex Keys

Most robots will have at least a few fasteners in them that need



a hex key. Keep a nice set together so you don't spend half your repair time searching for the right sized key. When possible, I prefer to use T handle style hex keys, however, a good set of L shaped keys are worth having because there will likely be occasions where you'll need the shorter profile and higher torque of the L keys.

In addition to this, many of the readily available power switches use hex keys for

operation. For these, I suggest long T handle wrenches. If possible, add a layer of heat shrink tubing to the majority of the key length to insulate it from the chassis and differentiate it from the rest of your keys.

In my toolbox, I have a set of both inch and metric Wiha ErgoStar keys. They're a bit pricy as far as hex keys go, but they're a good quality steel, and the case design makes it much easier to remove just the key you want.

Locking Pliers

A good set of locking pliers can be fairly versatile at an event. When paired with a vise or immovable slotted object, they can quickly aid in the de-bending of damaged armor. If you strip the head of a screw, a few flats ground into it and you can use these to break it loose. If you're hammering something flat, these move your fingers further from the hammer.

For such a basic tool, they're quite versatile and won't take up much space in the toolbox. There are a ton of options on the market to choose from. A traditional curved jaw plier will probably serve you better than a needle-nose in most general applications, so I'd suggest starting there.

Multi Tool

Exactly what you get here depends on the model, but you'll normally get at least a set of needle-nose pliers, wire cutters, wire strippers, a knife, and a screwdriver. In most tasks, it's not as good as the tool it's replicating. However, it makes



Multi Tool
with pliers,
knife,
screwdriver,
wire cutter,
and wire
stripper.



**Small non-marring
hammer with
plastic and brass
faces.**

for a compact way of having a spare of each of them. I know I've used mine at least once at every event I've been to.

Small Hammer

A small hammer can do quite a lot in a small package. If you can find one, the ideal hammer for this purpose is fairly light, has interchangeable heads (often nylon, brass, or steel), and will normally be less than 12" long.

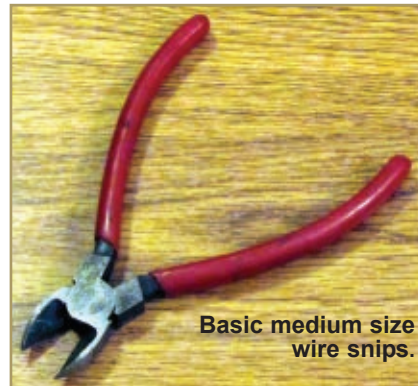
The softer faces on this hammer allow for quite a bit of energy transfer while not doing much — if any — damage to the surface of the material you're hitting. This is particularly useful when flattening bent armor panels or pounding down gouged areas. If you're searching for one of these online, a search that turns up quite a few results is "interchangeable head hammer."

Wire Strippers

It's tough to predict when you'll need them, but at some point you'll need a set of wire strippers in the pits



**Compact
wire
strippers.**



**Basic medium size
wire snips.**

at an event. If you don't already have a pair, it won't be expensive to get a decent one for your toolbox.

Wire Snips

While having wire snips and strippers may seem redundant, often wire strippers have the cutter deep within the jaws of the stripper. The snips are only meant for cutting. This means if you need to quickly extract some piece of electronics, you won't be stuck trying to cut it out with a knife or awkwardly fumbling with wire strippers. These are also quite handy for quickly removing zip ties if you use them to keep the wiring in your machine in order.

Adjustable Wrench

If you can fit a full wrench set in your box, go for it. If not, an adjustable wrench is almost as good and takes up far less space. Even if you've managed to not use nuts anywhere on your machine, it can still come in handy as an additional de-bending or prying tool.

Soldering Iron

If you have to do much of any electrical repair at an event, you're almost certainly sunk without one of these.

Rotary Tool

There's a good chance you've got one of these already, so you know how useful they are. The abrasive disks allow for shafts to be quickly cut down, lightening slots to be cut, stripped screws to be slotted, and

Mid-sized sliding clamp with rubber jaw covers.



wedges to be sharpened.

Sliding Clamps

Whether you're using them to hold parts in place while you're debending them, using them as a weapon lock on a small robot (do test to make sure it will actually stop the weapon before assuming it's capable), or just using them as an extra hand in the pits, these things always come in handy. If you can fit it, pack a few in several sizes.

1/4" Bit Set and Driver

A good bit set will make sure you've almost always got the screwdriver head on hand that you need. If you use the case, you'll have them all relatively organized in one spot, so you won't waste time digging when you need them.

Duct Tape

It's almost never the right tool for the job, but it will often get the job done for a while. If you forgot to make safety covers for sharp parts on your robot, some cardboard and duct tape will normally do it; if you've stripped out some screws on your armor, it can help keep it in place; and if you've got an access hole for a removable link, it can help keep the link from popping out of your chassis.

Luxury Items

If you have more room for tools than just a small toolbox, there are quite a few handy things that you can bring with you. In general, these will allow for faster modifications or

repairs during an event.

Cordless Drill

The big use of a cordless drill at an event is removing weight at the last minute. It's almost inevitable that someone will have finished their robot at the last minute and won't have gotten around to checking the weight until the day of the event. Whether it's you or someone else in this position, having a drill handy is a decent method of dealing with the problem.

Cordless Impact Driver

If you manage to make it toward the end of the bracket at an event, time starts getting quite tight. An impact driver combined with a bit set can dramatically speed up the removal and reinstallation of access panels between matches.

Angle Grinder

If you absolutely need to grind it flat, cut it off, or shave it down now, an angle grinder is the tool to use. You can get a wide range of wheels for them with grinding, cutting, or sanding as the primary focus. When

Angle grinder with sanding wheel.



used properly, a grinder can do the job a rotary tool would do in a fraction of the time. Just be careful with where the sparks or dust gets sprayed. Whenever possible, make sure to cover any openings into your machine with tape (or similar) to reduce the chances of metal shavings getting in and shorting something.

Large Hammer

As with the angle grinder, a large hammer will often do the same job as

the similar small tool, but much faster. A large sledge or demolition hammer should be a tool of last resort, but when it's needed to fix the problem, there is no substitute.

Bench-top Shop Tools

Occasionally at an event, a team will bring a drill press or other similar shop tool. If you've got the capability to bring one and are willing to allow other teams to use it as well, you'll rarely be alone at an event.

Drill presses, bench top mills, and small lathes all allow for a fairly extreme amount of damage to be repaired at an event, often by frantically remaking the damaged part.

Welder

If you've welded a portion of your machine together and you're able to bring your welder, it's worth the hassle. You'll be quickly able to patch up damaged sections or repair broken welds which can often be the difference between doing the repair correctly or using duct tape.

There are plenty of other tools that you can bring, but outside of necessary specialized tools, these are the first ones you should consider bringing to an event. While it's highly difficult to plan for every issue, these tools should have you prepared for the majority of situations you'll run into at an event.

There will almost certainly be something that has been left off this list that will be useful to you. As a final step of the packing process, I recommend taking the time to walk around your work area. Take a look at each of your tools and think about if it could come in handy during the event.

If you think you'll need something and you've got the space for it, bring it. Even if you don't end up using it, there's a good chance someone else will need one and your help could be the difference between someone making their next match or having to forfeit. **SV**

Rob Games Requiem: A Facebook Epic

● by the Robot Combat Community
as told by Kevin "Legendary Robotics" Berry

The announcement that RoboGames was not being continued by Dave Calkins and Simone Davalos started a FaceBook conversation with over 1,000 comments. A spinoff post worthy of recording as part of our "History of Robot Combat" series also ensued.

Jonathan Gilbert started it off:

Things I Will Miss About RoboGames:

- All the people: Builders, staff, medics, arena wranglers. Even Dave.
- 10:30 am safety meetings.
- Steel-toed boots.
- " !@#\$\$%^&* Calkins!"
- Using the forklifts.
- Ironed hot dogs.
- Ft. Mason on a clear day, in between matches, when you can look out past the dockside fishers and see Alcatraz.
- "Power off, and save your weapon."
- "The first transport is away!"
- Radio checks.
- The finest event/arena staff you could ever ask for.
- Medics that really don't want you to die.
- John Thorne and his hammer.
- Mike Thomas and his lack of fear of heights.
- Simone's ability to not take s*** from anyone.
- Mike Strange's sarcastic everything.
- Rob Purdy and his ability to do ANYTHING.
- Doing the thing with Heather, and doing it well together.
- That first really good hit against the arena — the first one of the event, when you look up at the shaking and the settling and you know that yup, she's got at least one more fight weekend in her.

Things I Will NOT Miss About RoboGames:

- 10:30 am safety meetings that follow a night of drinking that followed a day of construction.
- The arena roof.

- Unbolting the arena floor at 9 pm the night we're supposed to load out.
- Loading out in the rain.
- Ratchet straps.
- " !@#\$\$%^&* Calkins!"
- "Jesus, get your hands off your controller!"
- "No, I don't know if the schedule is current."
- "No, you can't go into the arena to just look."
- "Guys, you need to get off the ramps."
- "What do you mean, your controller's off? The bot is still moving."
- "How did you pass safety with that?"
- LiPo fires.
- Brazilians loudly banging on the glass.
- Last Rites.



AndyMark
Inspiring Mobility

**Specializing in Unique Wheels,
Gearboxes, Aluminum Sprockets
and Drive Bases**



8" Plastic Omni Wheel



c-Rio-Ready Full Chassis Kit



Aluminum Sprockets



6" HD Mecanum Wheel Set

AndyMark, Inc.
sales@andymark.com

Toll Free: 877-868-4770

www.andymark.com

- Running the pneumatics, only to have to re-run them after the compressor was moved.
- Walking around, finding sheared bolts.
- The daily mystery bruise count.
- The toilet paper at the San Mateo Event Center.
- Missing all the other events.
- Arguing the unstick rule.
- *Implementing* the unstick rule.
- Irregular bowel movements thanks to event food.
- Not seeing my apartment or daughter for over a week.
- Eating In-n-Out three out of six nights.
- 8' x 4' x 0.5 in Lexan sheets that are way too floppy to carry on your own.
- San Mateo's best bar to get roofied at.
- Getting a face full of builder @\$!\$ while on combat control.
- Union guys outside "encouraging" us to use them for labor.
- People who don't cover their weapons in the pits.
- People who don't cover their plumber cracks in the pits.
- Smelling like a goat in a chemical fire for three days.
- Daily bolts found in jeans count.
- All seven different sizes of fasteners for the arena.
- Needing to pay special attention to the dirt encrusted in your elbows. (Seriously, they're filthy. What would your mother say?)
- The ramp stackers.
- The giraffe dangles.
- The bad coffee.
- Load-in.
- Tear-down.
- Packing six pairs of jeans for three days of event.
- Industrial-sized bottles of Ibuprofen.
- The bananas.
- That first really good hit against the arena; the first one of the event, when you look up at the shaking and the settling and you ask why in all that is holy are you doing this again?

Much hilarity ensued; lots of it not printable. However, there was one thing that needed clarifying ...

Kevin Berry:

Okay, having never made it to RoboGames but having steeped myself in forum



and FB posts, I only have one question. Giraffe dangles?

David Calkins: Here's the 'giraffe dangler' story. It isn't as funny written down as it is orally, but here goes:

During the course of events, Simone and I often mutter to ourselves about what's got to be done. In the mornings, as we're showering, etc., a fly on the wall might hear entirely random comments such as "That's not gonna happen," "Gotta make sure the painting gets done," "Two jugs of coffee," or whatever. The comments aren't directed at anyone; they're just the normal busy-person-thinking-out-loud stuff.

So, one morning several years ago among the usually ignorable internal dialog, I hear Simone mutter what I thought was 'dangle a giraffe' I spun to her and said 'You're going to dangle a giraffe?' as I had this perfect vision of her standing on the raised combat control platform, holding an upside down baby giraffe by its hoof, screaming at the builders (sleep deprivation does funny things to your imagination).

What she actually mumbled was 'day-glo giraffe' (bright orange) as a reminder to herself about what the day's audience wristbands would be. Wristbands come in different colors and print patterns — such as striped, diamonds, or a 'giraffe' camouflage pattern.

She, of course, giggled at the mondegreen, and for the rest of the event I would call her a giraffe dangler. Somehow, the combat safety staff picked up on this, and since that event, giraffe dangler has been used by the staff as a generalized insult. Although, I personally will use the phrase 'stop dangling giraffes and get back to work' as a synonym for time-wasting.

Kevin Berry: Thanks for that. And, by the way, "10" on style. I'm been a professional writer and editor for 20 years, and I've never heard 'mondegreen' used correctly in a sentence. **SV**

Photo courtesy of
www.teamcosmos.com.

Photo courtesy of
www.teamkiss.com.



EVENT REPORT:

Robots Battle at the County Fair

● by Andrea Suarez

The Miami-Dade County Fair & Expo brings thrilling rides and mouth-watering food to South Florida each year. Kids eagerly anticipate the event, and submit their entries for the various categories of exposition including arts, poetry, agriculture, architecture, and engineering, among many others. My favorite, of course, is the Robotics exhibit and demonstration, which lured in the crowd during the 15 lb combat competition. Spectators looked on with excitement as the high school competitors provided electrifying matches in the arena and frantic repairs in the pits.

In an early first round match-up, Christopher Columbus High School's Black Hawk attacked Ransom Everglades' Unicorn with its spinning vertical disks, but Unicorn used its unicorn-like bar to corner Black Hawk and pin it against the wall.

After failure of its weapon belt, Black Hawk took advantage of its speedy drive motors to spin in place and get some hits on Unicorn as it slammed it against the wall to win the match.

Miami Lakes Education Center's all-girl team competed with Sting'em, which is a powerful spinning drum with a brushless drive system. They won their first match against their schoolmate's powered wedge lifter, Mr. T. Also from the same school, Rattler 2 used its spinning drum to fight Ransom Everglades' Assault and Battery.

Twin brothers built this multi-bot, made up of an 8 lb robot with two vertical disks (Battery), and Assault — a 7 lb robot that uses a vacuum to behave as a heavier robot.

During their first match, however, a last minute problem with Assault forced Battery to fight the match on its own. Astonishingly, Battery was able to dominate Rattler 2 once its weapon was inactive, and Battery won the match at only half the weight of its opponent.

I was glad to see that Team StarBot brought Pachanga — a pneumatic flipper — to the competition. We see these designs less frequently as spinning weapons get more powerful, but this team proved that a well-designed flipper

can still be a very dangerous opponent.

The team competed with their robot at last year's STEM Tech Olympiad, and they have continued to improve their design. Their first match against MLEC's Sting WMD was one of the closest matches of the event. Pachanga got plenty of flips on Sting WMD early in the match, but Sting WMD was able to get some impressive hits and push Pachanga around to dominate the second half of the match. Avoiding a very difficult judge's decision, Pachanga suddenly stopped moving with less than 30 seconds left on the clock!

Christopher Columbus High School's Black Night was another attention-grabbing design. This robot has a long wedge leading to a drum weapon that aims to launch the other robot into the air. To achieve this, their drum weapon spins in the opposite direction of traditional drums.

Ransom Everglades' La Caja took advantage of this and rammed their box-like robot into the back of Black Night's drum. Since the drum

Assault's battery gets ripped out of the robot!



Robotics exhibit draws a crowd!



Pachanga's pneumatic flipper gets a good lift!



Unicorn needs some quick repairs.



Long-time rivals Ransom Everglades and MLEC.



Assault & Battery (multi-bot) team up against La Caja.



spins down, this sent Black Night flipping through the air and slamming hard enough into the floor that a drive motor fell out of the robot!

In the semi-finals match, Pachanga's flipper got stuck deep under one of the floor boards, but some frenzied transmitter operation broke the robot free as their luck turned around. Rattler 1's brushless drum delivered some

massive hits, but its weapon belt came loose and gave Pachanga a chance to seize the win and advance to the final match against the undefeated La Caja.

In this last quick-paced match, Pachanga got a few lifts as La Caja tried to push their robot around, but their luck ran out suddenly when the rubber fell off their drive wheels and rendered their robot immobile in the last minute of the match. This concluded the day of fighting, with Pachanga finishing second and La Caja taking home the first place trophy.

The judge's awards are often the most coveted of the competition. This year's "Most Creative" award went to Ransom Everglades' Assault and Battery for their distinctive multi-bot design. Pachanga won the "Best Engineered"

award for its effective flipper design, which is often hard to implement well.

The Youth Fair requires each team to submit documentation with their robot to develop these skills that are so often required in science and engineering careers. The Miami Lakes Education Center robots put on a show of sparks with their drum bots as they advanced

through the brackets, but they also swept the documentation awards with their well-designed robots!

Rattler 1 and Sting'em took third and second place, respectively, and Sting WMD won the first place documentation award.

It was an exhilarating day of robot battles for both the competitors and the spectators. Most of the audience had never seen a combat robotics match before, and many of the kids were motivated to get involved.

The event was organized by Miami-Dade County Public Schools and StarBot — which is a non-profit robotics shop in Miami, FL that has given kids of all ages the opportunity to get involved in robotics for over 15 years (www.starbotinc.com). A big thank you to Paul Kynerd, Coach David Kirkpatrick, Richard Wong, and Nola and Bill Garcia for making the event possible.

Nola summarized the day, "Sunday's battling robot event was a crowd pleaser as robots were thrown up in the air and sparks flew! Many young spectators were inspired as they watched with eyes wide open! The students who built and drove the robots should be congratulated as their hard work was evident in the great designs and strength of the robots that competed. These truly will be our future technological leaders!" **SV**



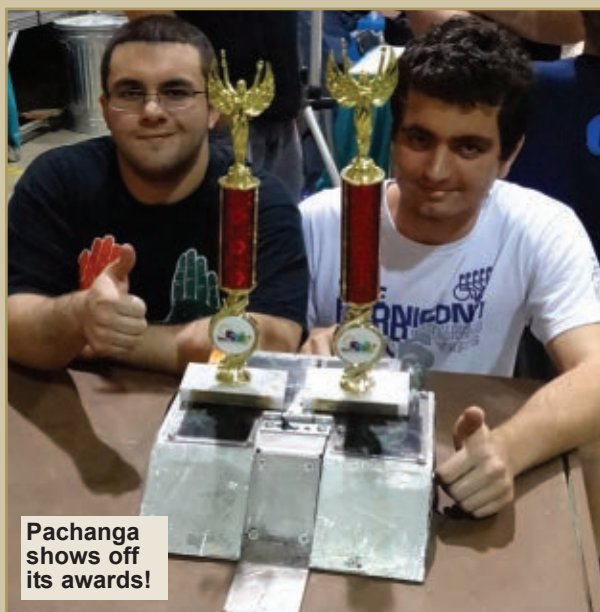
MLEC's all-female team receives their award for documentation.



Black Night loses a drive motor!



Unicorn after a tough loss!



Pachanga shows off its awards!



MLEC works on their powerful drum spinner in between matches.

PRODUCT REVIEW: Snap Hubs

● by Pete Smith

Kitbots and FingerTech Robotics have joined forces to develop a new range of hubs for smaller bots.

Snap Hubs are the result of years of combat experience and offer high performance, problem-free tire and hub replacement, all at a lower cost and weight.

The tire is retained by a custom washer that is held in place with a snap ring. The washer features a recess that locks the snap ring in position, reducing the chance that it is damaged in combat.

The hub is secured to the motor shaft by a larger (8-32) set screw. This allows a larger and more robust hex key to be used, and reduces the chance of the set screw recess or hex key rounding off.

The hubs are anodized for good looks and come in two sizes to suit the popular 3/4" wide Lite Flite or the narrower 1/2" wide Lectra Lite



Tires, with 3 mm or 4 mm bores.

The wide hubs each only weigh 0.25 oz (7g) complete; the narrow ones are only 0.21 oz (6g). **SV**

Kitbots
www.kitbots.com

Fingertech Robotics
www.fingertechrobotics.com

Melty Brains

Tales of RoboGames

by Kevin Berry

Annual Meeting
Professional Philological
Practitioners

This year featuring
Polemical Contrivances!

Wha??????

In the rubicund equilateral
quadrilateral : *"The Colometry of
Finglass and Sophocles' Manuscript L"*

And in the
cerulean quadrate
lineation:
*"Rewriting the
Demiurge: Galen's
Synopsis of
Timaeus and Ex
Nihilo Creation"*



Next year we'll try the
Lexical Functional Grammar
Conference

If you do, I'm SO
naming my bot
*"Clitic Placement,
Syntactic
Discontinuity, and
Information
Structure Fist of
Death"*

*Desperate for a sponsor, the new RoboGames organizers go in an
unfortunate direction ...*

Tips From the Pits:

Getting Connected

● by Pete Smith

The “magic smoke” from burned out motors and electronic speed controllers (ESCs) is inevitable in combat robotics, and getting those parts replaced quickly can make the difference between working your way through the brackets to the finals and an ignominious forfeit and early exit.

It's not always immediately obvious what has failed ... is it the drive or weapon motor that has failed? Has the ESC or battery failed, or is it simply a loose connection?

The first thing to check is the last possibility ... a broken solder joint, a failed crimp, or simply a connector that has come unplugged. Any of these can bring a bot to a halt just as effectively as a failed component. As a rule, good soldering, heat shrink tubing, duct tape, and tie wraps can prevent or at least reduce these problems to a minimum. It's a lot quicker than replacing parts, so unless the failure is very obvious, take the time to use these things.

One might be tempted to avoid connectors and simply solder all the connections together, but resist that urge! It makes problem identification and part replacement much slower. Fit connectors between each component.

Use polarized connectors on the power input side of ESCs since connecting these up the wrong way will usually blow up your replacement part.

I use Deans type or XT60s on my 12 lb and 30 lb bots because they provide high current capability, solid polarization, and good retention. The Deans have the edge

on size over the XT60s, but the XT60s are easier to solder.

Powerpoles are easy to use, but are relatively bulky. The contacts can sometimes get bent and fail to make good contact, and retention is not the best (I always add a tie wrap.)

Until recently, I used JST connectors on my Ant and Beetleweights, but now I use the new Mini Black Ts. The JSTs require a special crimping tool that takes a lot of practice to use quickly, plus the polarizing is not very reliable. I smoked a couple of ESCs before learning to be very careful when using them. The Mini Black Ts are compact, solidly polarized, and have good retention.

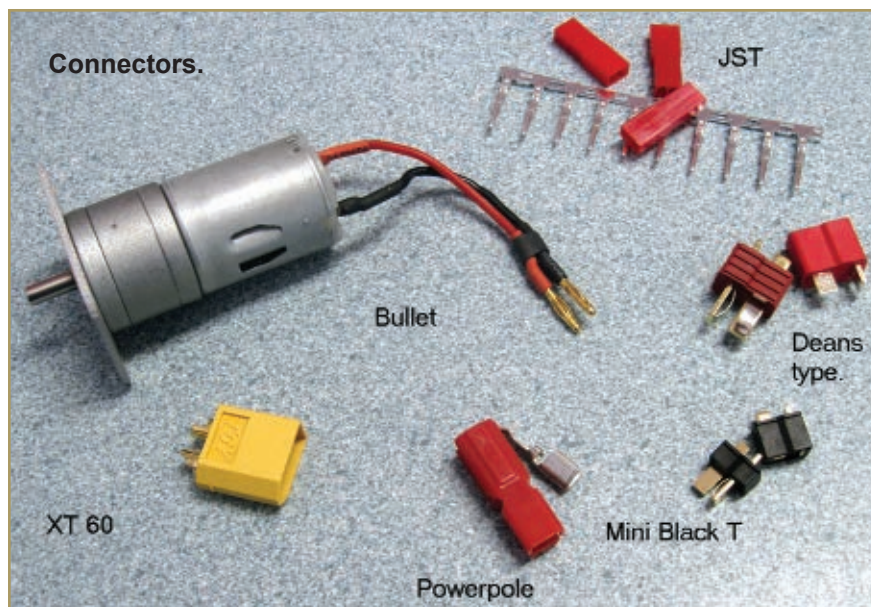
On the output side of an ESC or on motors, I like to use bullet connectors. These are standard on the brushless motors most folks use now for weapon motors, but they are useful on DC drive motors as

well. They allow you to quickly attach a new motor to see if it's the motor or the ESC that's failed. Because they are not polarized, you can swap the leads around to reverse the rotation of the motor if it's turning the wheel in the wrong direction.

With all connectors, remember to use the female ones on the output wires as these can be live. The female sides make it less likely that you will get a shortcircuit.

If you change connectors on the batteries you use (many come with the JST as standard), change over one lead at a time carefully insulating any exposed metal. Even a small battery pack contains a surprising amount of energy.

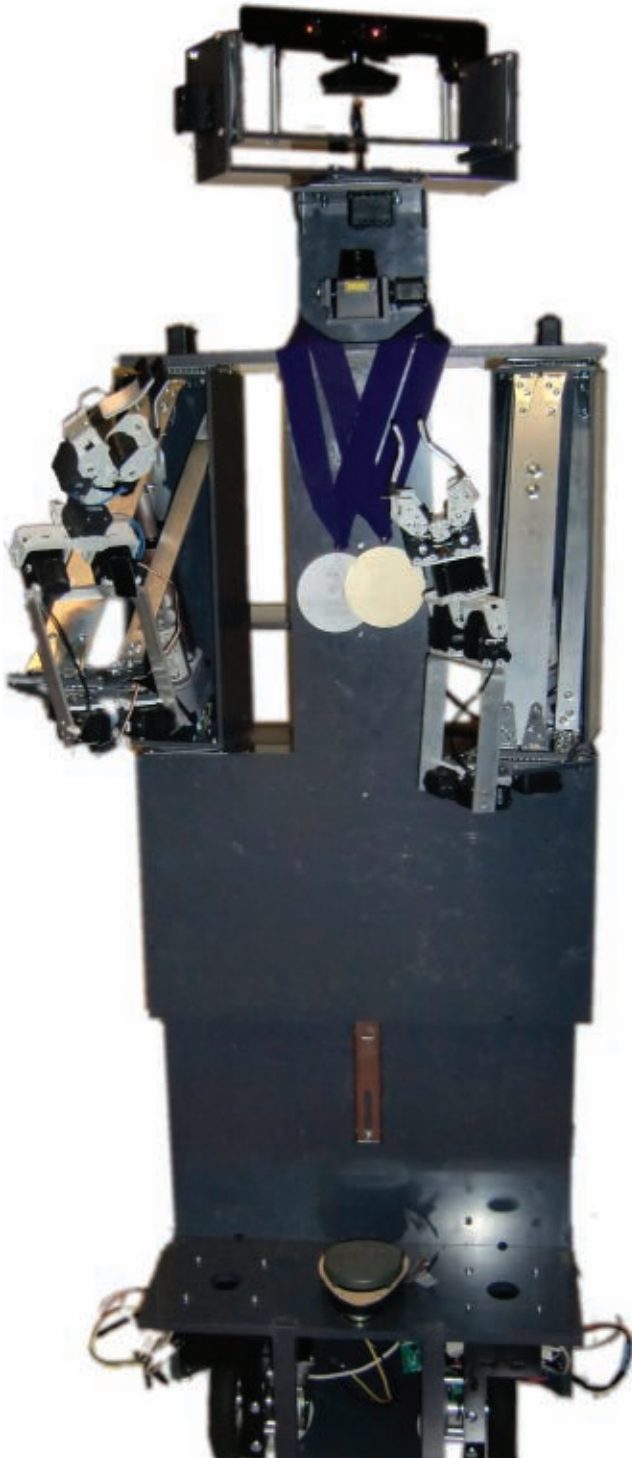
Use the right connectors and you can have your bot up and running again while your rival is burning their fingers on a hot soldering iron replacing the wrong part! **SV**



PR2Lite Grows Up — and So Do Its Makers

by Alan Downing, Matthew Downing, and Frank Ou

You can discuss this topic at <http://forum.servomagazine.com>.



PRLite was first featured on the cover of the December 2011 *SERVO Magazine* (**Figure 1**). In that article, we described how four high school students and their mentor fathers prototyped their own PR2. PRLite had its own tilting LiDAR, a Kinect, a 12" torso lift, two Bioloid arms, a Neato LiDAR on its base, four partially pivoting Parallax motors and wheels, and more. However, PRLite had a short wooden frame with arms positioned at the bottom of the front torso. It didn't look much like Willow Garage's PR2.

Much has changed over the last two years. First, three of the builders are now in universities and make only occasional appearances at our weekend meetings. Similarly, our robot has grown up to look and act more like a PR2. Now made of machine-cut PVC, PR2Lite is between 40" and 52" high at the shoulders, and has LiDARs and a Kinect placed similarly to their PR2 counterparts. Plus, PR2Lite sports new upper arms powered by linear actuators, a new CHR-UM6 IMU, and a lot of new ROS-compatible software.

Our PR2Lite team now mostly consists of Alan (software), Frank (hardware, networking, firmware), and Matthew (builder, designer, high school robot club president-elect). **Figure 2** contrasts the old wooden prototype with the revised PR2Lite.

Design of the Body

Matthew designed and built PR2Lite's body. He used Autodesk Inventor to model the pieces which were then machine-cut from quarter and half inch PVC by Mr. Plastics in Oakland, CA. Like PR2, our robot is the appropriate height to manipulate objects on tables and do other human interactions. The Autodesk Inventor screenshots show the design of the arms and upper torso when lowered and raised (**Figure 3**).

The upper arms are lifted by 4" linear actuators and rotated at the shoulders by two AX-12 servos. The linear actuators are

kept close to the body and ROS-compatible software converts the linear length to angle, and vice versa for position feedback. The upper arms use a parallelogram design to allow the lower arm to remain on a horizontal plane (**Figure 3**).

Four inch wide lazy susans on top and below the compartment for the linear actuator make it easier for the servos to pan the arms. The lower arms are currently similar to AX-12 CrustCrawler arms with dual grippers.

The base is 9" high — the same as PR2's base — and can hold up to five large 12-volt sealed lead-acid batteries. The wheel design and software was derived directly from our original PRLite prototype. The computer and electronics are largely contained throughout the torso.

The Autodesk design was then manually converted into a URDF model for ROS to do the different transforms, display, and planning for PR2Lite. The passive cyclic joints are a problem for ROS' URDF which requires an acyclic tree-structure.

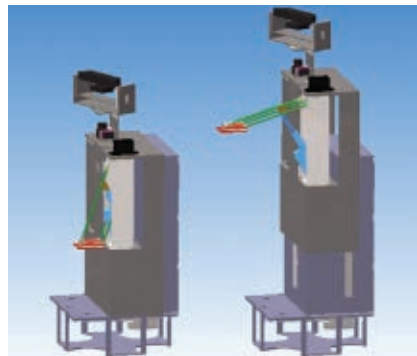


FIGURE 3. The robot has a telescopic body, lifting up by one foot. Also notice the parallelogram shape of the upper arms (green) which keeps the elbows (red) horizontal — even when the upper arms lift up.



FIGURE 1. The December 2011 *SERVO Magazine* with a picture of the first PRLite prototype beside Willow Garage's PR2.

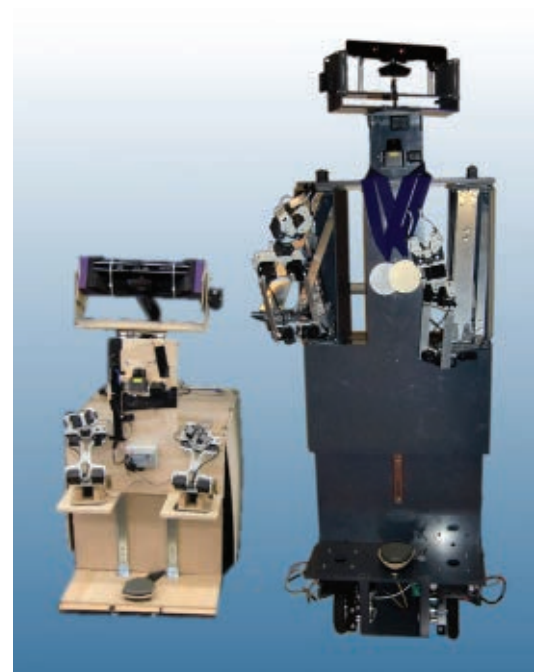


FIGURE 2. The first prototype (left) beside the new and improved PVC PR2Lite (right).

For example, PR2Lite's panning shoulder, the arm, and the linear actuator form a triangle with two passive joints.

The linear actuator is along the hypotenuse and raises or lowers the arm when the actuator is

extended/retracted. In our URDF, we artificially break the cycle by fooling ROS into thinking that the complex shoulder joints are just another Dynamixel joint.

Basic trigonometry translates the linear length into an angle of rotation for the upper arm that is returned in a Dynamixel joint state message.

The URDF is used by ROS to do transformations between different frames and joints, allowing general control of robots with many degrees of freedom. An output of *roslaunch tf view_frames* gives an idea of the scale of PR2Lite (**Figure 4**).

New Linear Actuator and Wheel Controllers

Before leaving for Berkeley, Robert redesigned PR2Lite's controllers for the linear actuators and wheels. His firmware implemented a protocol stack over RS-485 that broadcasted datagrams or stream-based connections with packet collision avoidance and retry capability. The RS-485 provides much improved electrical noise immunity compared to the I²C bus in the old PRLite.

The new design also uses MOSFETs to replace the old relays. The wheel controllers utilize photogates to act as encoders and provide PID control. Unavailable at the time, Parallax has now changed the design to their wheel controllers to also support this functionality.

Although the microcontrollers on the RS-485 bus can try to avoid packet collisions, it is hard for the PC with the USB serial port to do so. Frank has changed the software to give priority to the

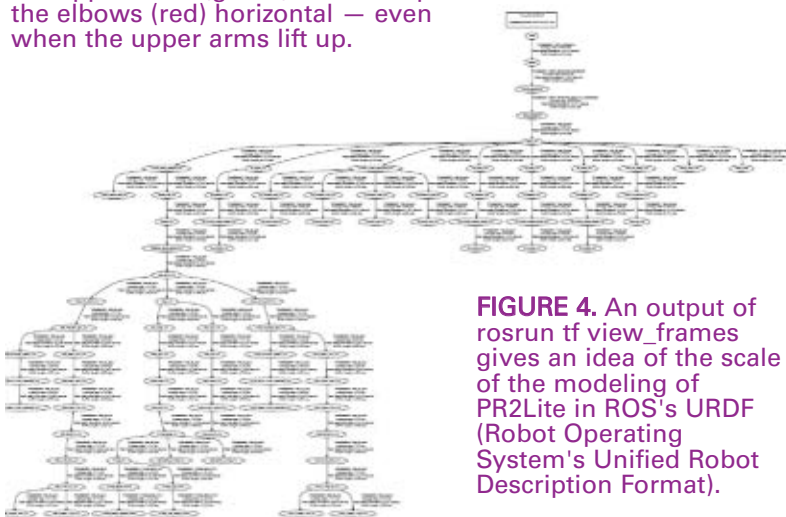


FIGURE 4. An output of *roslaunch tf view_frames* gives an idea of the scale of the modeling of PR2Lite in ROS's URDF (Robot Operating System's Unified Robot Description Format).

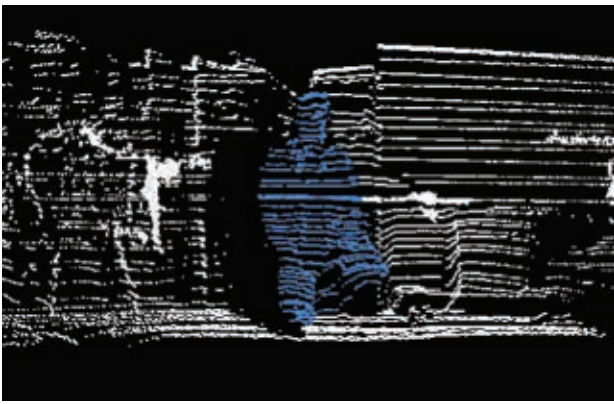


FIGURE 5. A kneeling person seen in a PointCloud from PR2Lite's tilting LiDAR. The person has been highlighted in blue in post-editing to be more obvious in the picture, but is actually very easy to see in the 3D model created by the PointCloud.

PC to transmit packages over the RS-485 bus. All the microcontrollers will withhold from transmitting packages when the PC is talking.

This method reduces packet collisions between microcontrollers and the PC, and still allows the microcontrollers to send out frequent position updates from the linear actuators and wheels.

The four wheels of the PR2Lite are driven by four motors with four wheel controllers, and must start and stop at the same time. A cable connecting the four wheel controllers was added to function as a synchronization signaling bus to ensure that all four wheels will move at the same time.

For example, if for any reason one wheel controller fails to receive a move command from the PC, the other three wheel controllers will not start. By monitoring the status of all the wheel controllers, the PC can decide to retransmit the move command or to cancel the move command. This mechanism ensures that PR2Lite will not crash due to communication errors with the wheels.

Tilting LiDAR

PR2Lite has a Hokuyo URG laser distance center that can be tilted by a Dynamixel AX-12+ servo. Like PR2's tilting LiDAR, ours is located on the neck below the 3D camera. We modified the University of Arizona (UoA) Wubble2 code to set tilting speeds separately and to publish transforms. The UA demo program also assembles scans and publishes a *point_cloud*. The current assembler doesn't self-filter or shadow-filter.

Eventually, we'll convert the PointCloud to PointCloud2, and then use the PCL library. In addition, the LiDAR will also be statically tilted for obstacle avoidance in SLAM. **Figure 5** shows the LiDAR's PointCloud of a chair and kneeling person.

Neato LiDAR

We disassembled a used Neato and extracted its motherboard and LiDAR. The Neato LiDAR is the main

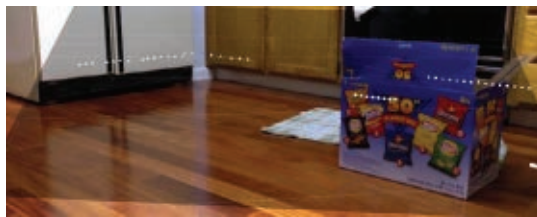


FIGURE 6. Notice the points that the Neato sees superimposed over the kitchen it was looking at.

sensor for SLAM. Like with PR2, the LiDAR was positioned horizontally on the base of PR2Lite, and the motherboard was placed on a shelf within the torso. We replaced the original motherboard's instantaneous "ON" switch with an LED so that the LiDAR will turn on whenever the ROS node is started.

We kept the whole motherboard for ease of development. At some time in the future, we may also use the Neato's sonar and edge detection sensors. The batteries must be attached for the motherboard to boot, even though we have the AC recharger permanently attached to the motherboard. The AC recharger can be powered by a battery through a DC to AC converter.

The Neato is also used by SLAM. PR2Lite's SLAM has been configured to use a new CH Robotics IMU and the latest revised PID. The CH Robotics IMU was chosen because it already has an ROS-compatible driver and firmware.

Figure 6 is picture of our kitchen with the LiDAR output superimposed. Soon, similar low-end LiDARs are expected to be for sale — without requiring purchase of a used vacuum.

Arm Navigation

Our work on the PR2Lite arm navigation is built upon the non-PR2 ROS pioneers, specifically Pi Robot (Patrick Goebel), Maxwell (Michael Ferguson), Turtlebot (Willow Garage), and Wubble (Anton Rebgun) — all of whom configured ROS to handle Dynamixel-based arms.

Unfortunately, there was no perfect precedence for us:

- The Turtlebot and Maxwell use the simple arm driver with Arbotix.
- Wubble uses OpenRave for the trajectory planning and wrote a custom *wubble_follow_joint_trajectory*.
- Pi Robot uses OpenRave for the trajectory planning with USB2Dynamixel.

The issues that are different for PR2Lite include:

- A passive joint formed by a parallelogram for the upper arm that kept the elbow pan joint level with the ground as it is raised or lowered. ROS does not have built-in support for URDFs with joints that are not acyclic.
- Having two complex arms.
- Having a linear actuator to lift the upper arm. The linear actuator is part of the closed-loop kinematic joint chain not supported by ROS Electric arm navigation.
- Using the ROS Electric arm trajectory planning code, along with a USB2Dynamixel.

We updated PR2's Arm Kinematics plug-in to hard-code our passive elbow joints based on the shoulder angle of the upper arm. Simple changes were made so that the IK seed state input satisfies our parallelogram constraint.

Once we got the parallelogram arms supported, the Arm Navigation Warehouse tutorial worked great in simulation. Getting the arm navigation to work on the real robot was much harder. Our configuration to execute arm navigation on a real robot was derived from the following:

- The Dynamixel controller by Anton Rebgun (UofA) which evolved from the simple arm controller.
- The Dynamixel joint state publisher from Pi Robot which maps the Dynamixel controller to the joint names, and then puts them into an array of *joint_states* as required by the *joint_states* message. To make coding easier, we implemented a naming convention for the joints, controllers, and left/right sides.
- The Arbotix *follow_joint_trajectory*. This is the key missing piece that maps the ROS electric arm navigation to the Dynamixel state/commands. The baud rate and read/write characteristics for the USB2Dynamixel have to be set correctly. To satisfy the precise arm movements required for chess, for example, we extensively rewrote this node for PR2Lite's configuration to check for joint status to ensure precise following of trajectories while compensating for motor stalls.
- The *fake_pos.py* from Turtlebot. The warehouse viewer will hang if the non-fixed joints in the URDF are not sending out fake positions. To get the list of missing joints, use *rxconsole* and *rxloggerlevel* to obtain the DEBUG messages for the environment server.
- A transform publisher to send out the quaternion for the robot.
- The launch file for the ROS planning scene warehouse visualizer required not using *sim_time* or *fake_time*; setting *use_monitor* to true; setting *execute_left_trajectory*; and setting *use_robot_data* to true.

Each of the above requires their own configuration that needs to be consistent with all of the others. If one thing is not perfect, the ROS warehouse viewer will hang and require debugging using *rxgraph*, *rxconsole*, *rxloggerlevel*, *roswtf*, the *tf* tools, etc. It's not easy and there is a learning curve.

Once the arm planning worked in the visualizer, for the real robot, the arm planning was still not effective enough for PR2Lite to play chess. The generic arm planner in *arm_navigation* generally failed to find a valid plan due to a lack of a good analytic solver.

For chess, we had to make many incremental plans using the numeric solver based on current or projected position. The wrist points the grippers straight down to



FIGURE 7. You won't be able to read the graph, but it gives you an idea of the complexity of the configuration of the arm navigation.

make the solutions simpler but, if necessary, it can rotate 360 degrees while still pointing downwards.

Movelt! is the next generation arm navigation software for ROS. It has been recently released and solves many of these final issues, but this early release of Movelt! introduces new ones. It may be time for us to upgrade. Up to now, we have deferred the upgrade from Electric to Groovy to avoid the very different build infrastructure and incompatible APIs.

The *rxgraph* for PR2Lite for both arm navigation and the base is shown in the graph in **Figure 7** where each of the nodes is a different package (written in C++ or Python) with messages being passed between them. You won't be able to read the graph, but it gives you an idea of the complexity of the configuration.

GUI

We modified a GUI by Patrick Goebel for Pi Robot that can be used both by tablets and laptops. **Figure 8** shows a sample screenshot. Further enhancements are planned.

References

<http://mattdowning.wordpress.com/pr2-lite/>

www.servomagazine.com/uploads/issue_downloads/PR-Lite.pdf

<https://github.com/rqou/prlite-pc>

www.willowgarage.com/velo2g?

www.pirobot.org/

www.ros.org/wiki/maxwell?

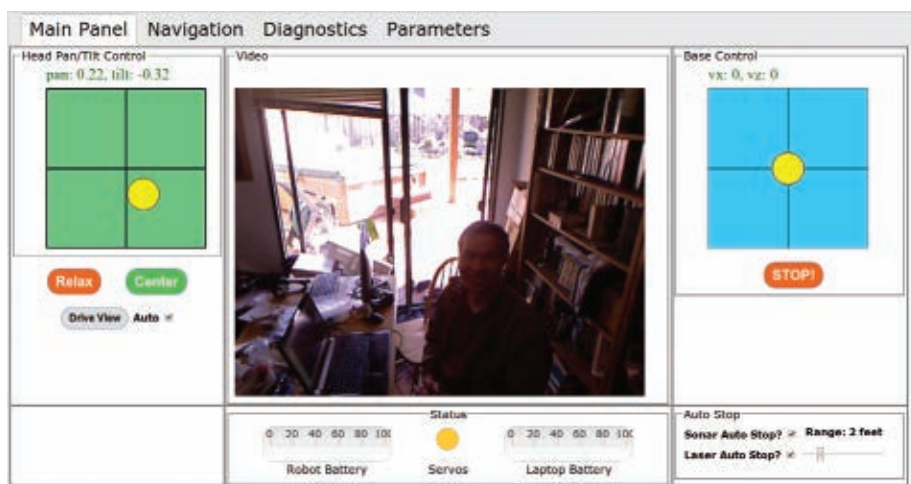


FIGURE 8. A screenshot of the GUI, meant to be compatible with tablets and laptops. Thanks Patrick Goebel (Pi Robot)!

Maker Faire 2013 Demonstration

At the 2013 Bay Area Maker Faire, PR2Lite took on Maxwell and Michael Ferguson in a robot vs. robot battle of chess (**Figure 9**). Michael and Maxwell are the 2011 champions of the AAAI small manipulator chess challenge.

Michael also wrote a series on Maxwell in the April, May, and June 2012 issues of *SERVO Magazine*. Both robots were running different variations of the same code.

The Kinect was used to identify where black or white pieces were located and whether the pieces were moved. GNUChess was used to implement the rules of chess, validate legal moves, and track which pieces were where. Finally, PR2Lite used ROS Electric-based arm navigation while Maxwell used the recently released MoveIt! for ROS Groovy.

Maxwell was a formidable opponent, and played an excellent game of chess. PR2Lite tried hard, and provided entertainment when it didn't work. One time, it accidentally picked up its king on its first move, and mistakenly resigned



FIGURE 9. PR2Lite (right) plays Michael Ferguson's Maxwell (left) in chess at the 2013 Bay Area Maker Faire in San Mateo, CA.

by dropping the king off the table. Another time, PR2Lite did its best to swipe off all its pieces due to the servos beginning to overheat. This showed that although the AX-12s used for the arms were sufficient for simple teleoperation, they were underpowered for the lengthy arm planning and the precision movements required for chess.

For Maker Faire, we also constructed a new transportation base. PR2Lite is large and heavy, and requires strong supports to prevent it from tipping or jostling in transport. The new transportation base can be strapped to the loops that secure the removable seats of a minivan.

PR2Lite's head needs to be detached to fit through the side door. A hydraulic motor cycle lift is used to safely raise it to the right height for the car door. In contrast, Michael's Maxwell can be easily disassembled and packed into a small suitcase.

Future Plans

PR2Lite continues to evolve. Powerful MX64 and MX106 servos and a Velo gripper courtesy of Willow Garage will soon replace the AX-12s in one of its arms (**Figure 10**). In addition, the arm planning will eventually use MoveIt!. To extend mobility time, we plan to replace its current battery-draining computer and its DC to AC converter with powerful laptops.

PR2Lite has now reached the level of maturity that applications developed for PR2 can be adapted for its own use. Prior PR2 hackathons have demonstrated keyboard playing and drawing capabilities that look like good potential initial candidates. Also, the new HBRC floorbot challenge is intriguing. Information and the latest updates are available at <http://mattdowning.wordpress.com/pr2-lite>.

Only five years ago, it was inconceivable that a hobbyist robot this advanced could be built mainly by high school students for price tag significantly lower than a Nao or Darwin humanoid robots. Now, you can build your own PR2 too! **SV**



FIGURE 10. A Velo gripper (left), courtesy of Willow Garage next to our current gripper (right). The Velo is already larger than the current gripper, and it will be even bigger and heavier with servos.



www.uat.edu/majors



Tech Respect

LEARN

Advancing Computer Science
Artificial Life Programming
Human-Computer Interaction
Open Source Technologies
Robotics and Embedded Systems
Strategic Technology Development
Technology Product Design

Digital Media
Digital Video
Enterprise Software Development
Game Art and Animation
Game Design
Game Programming
Network Engineering

Network Security
Serious Game and Simulation
Technology Forensics
Technology Studies
Virtual Modeling and Design
Web and Social Media Technologies

Bachelors | Masters | Online | On-Campus | www.uat.edu

Please see www.uat.edu/fastfacts for the latest information about degree program performance, placement and costs.

Making Better Arduino Robots

By Gordon McComb

FIGURE 1. The completed ArdBot II is a two-deck expandable desktop robot using an Arduino Uno microcontroller board.

with the ArdBot II

Part 1

Back in November 2010, I wrote a multi-part article for *SERVO* on the ArdBot — an inexpensive desktop robot made for expansion. Its main goal in life was to explore a room looking for things to bash into, but hopefully avoid.

The concept of the ArdBot proved popular, and I went about improving the design with a more slim line version, plus adding new features along the way. Thus, the ArdBot II was born. In this month's article, I'll introduce the ArdBot II, and show you full construction plans so you can build your own.

In upcoming installments, I'll show you how to add not one, but several types of Arduino boards to the ArdBot II; how to operate it using an inexpensive infrared remote control; ways to endow it with numerous sensors — among them multi-sensor ultrasonic and laser range finder — plus more.

ArtBot II Design Concepts

Like its forerunner, the ArdBot II uses two standard size servo motors for propulsion. The motors are conventional R/C servos that have been modified to operate in continuous rotation. That is, instead of a regular servo that moves to a specific angular position, then stops, the motors on the ArdBot II continuously spin clockwise or counter-clockwise, depending on a control signal your Arduino emits.

You can get standard size R/C motors that have been designed for or modified at the factory for continuous rotation (CR), or hack the servos yourself. CR servos are now fairly common, thanks to the growing popularity of educational robots. Among my favorites is the Parallax continuous rotation servo. Other brands of CR servos include GWS and Spring RC.

The motors and their wheels are mounted on the sides of the bot. This makes the ArdBot II differentially steered, meaning that movement is dependent on the difference in

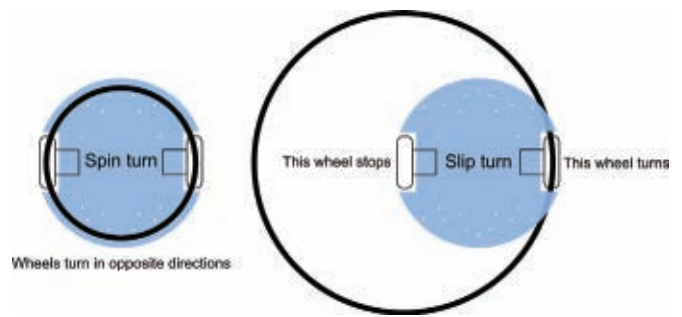


FIGURE 2. By controlling the servo motors, the robot can be made to travel in straight lines or make turns. By reversing the motors relative to one another, the robot will spin in place.

direction and speed of the motors. The robot can go forward and backwards, of course, as well as turn and spin in place. **Figure 2** shows various turning modes, depending on how the motors on each side are controlled. The robot can spin in place, for example, by reversing one motor relative to the other.

For balance, the ArdBot II uses simple skids as front and back casters. I chose static skids because they're small, easy to find, and best of all, cheap. The robot is lightweight, so there's no problem of causing scratches or other marks on floors.

It does mean, however, that the robot requires a fairly flat surface for traveling — carpets and kitchen floors are ideal, just as long as you stay away from trying to go over old socks, magazines, electrical cords, gravel, and other raised objects. There isn't enough clearance between the bottom of the robot and the ground to trounce over rough terrain.

The ArtBot II uses two decks for expandability, and an additional deck can be easily added if you need more mounting space. The bottom and top decks share the same dimensions: 5" by 7" inches. Because they are the heaviest components, motors and batteries are located on the bottom deck. This keeps the center of gravity of the ArdBot II low, where it should be. The Arduino and other electronics go on the top deck.

Arduino and Power

For the prototype ArdBot II, I've used a standard Arduino Uno board, mounted toward one end of the robot's top deck. A separate mini breadboard is stuck into place nearby. The breadboard serves as a convenient "patch panel" of sorts to connect the Arduino to the motors, sensors, and other parts.

Most active robotics components typically require at least three wire connections: power, ground, and signal. The Arduino has only one 5V power pin, and the arrangement of its other pins makes attaching to things like servo motors awkward, at best. The breadboard provides a quick and easy way to consolidate the wiring, allowing you to merely plug in the motor or other component.

I chose a separate Arduino breadboard shield mainly because of cost — the mini breadboard goes for about \$4 and attaches to the robot in seconds. The typical solderless

Other Battery Options

An alternative to using separate sets of batteries is to use one larger rechargeable pack. Whether or not you can go this route depends on the servo motors you use, or whether you want the extra hassle of adding a high current voltage regulator for the motors.

Off the shelf, most servo motors (being designed for use in radio control models with 4.8 volt power packs) are rated for up to six volts. Some brands and models can be successfully used at up to 7.2 volts, though the manufacturer may not specifically indicate it due to liability concerns. At higher voltages, the motor can get hotter, drawing more current and possibly damaging its electronics. Servo makers don't want to go about replacing their products, so they tend to be conservative with their ratings.

With this in mind and if you're sure the servo motors you use operate satisfactorily at 7.2 volts, you can power the Arduino from a single robust set of rechargeable cells. For 7.2 volts, you'll need six cells — $1.2 \times 6 = 7.2$ volts. You can either buy a six-cell pack ready-made or construct your own.

Note! Some digital servos are designed for 8.4 volts and higher. By digital, these servos incorporate digital circuitry in order to add special functionality such as speed control. Servos that are digital say so, and often cost considerably more. But take heed. Many types of digital servos are not adaptable for continuous rotation use. You're better off using the old fashioned — and less expensive — analog servo, such as the Parallax model mentioned towards the beginning of the article.

Here's the reason to avoid digital servos as the main drive motors in your robot: Servos are controlled by sending them pulses; the length of the pulses communicates its desired position to the motor. Longer pulses move the servo to a position one way, and shorter pulses move the servo the other way.

For robotics, the same technique is used but rather than specify a precise angular position, the duration of the pulses determines the direction the motors turn and their speed. In analog servos, stopping the pulses altogether — something you may have reason to do — causes the motor to coast to a stop. Without pulses, the motors are effectively depowered. In a typical digital servo, stopping the pulses may have no effect. Digital circuitry inside the servo "remembers" the last pulses it receives, and continues to apply those to the motor.

breadboard shield costs about \$15, and often requires soldering connector pins in order to electrically attach it to the Arduino. However, there's no rule that you *must* use a separate mini breadboard. If you prefer the breadboard shield, by all means use one. In a future installment, I'll show the ArdBot II with a unique third-party miniature Arduino board that is designed for easy connection to servo motors and common sensors, bypassing the need for a separate breadboarding area.

Power for the ArdBot II comes from a pair of battery packs. A set of four AA batteries provides operating juice

for the motors, while a separate nine volt cell gives power to the Arduino. Why two sets of batteries? While other power arrangements are certainly possible, this one ends up being among the least expensive and easiest to implement. The Arduino's built-in five volt regulator requires about 1.5 volts of overhead. That means the (nominal) six volts from four AA batteries may not provide enough voltage to adequately operate the Arduino.

I say "may not" because the output of the four-cell pack is not a predictable six volts. As the batteries wear down, the voltage drops — possibly to a level below the Arduino's voltage requirements. When that happens, the microcontroller may spontaneously reboot, restarting its programming. Such behavior can be maddening to deal with because it's not always clear when or why it's happening.

By running the Arduino on its own nine volt battery, there's plenty of voltage overhead to spare, and problems like spontaneous restarts are far less common. Even as the motor batteries wear down, the Arduino's power cell continues to deliver all the necessary electrons.

You can also use the two-pack battery feature to power just the Arduino without the motors. This is handy, for example, when you're experimenting with a new sensor and you want to keep the ArdBot II tethered to

FIGURE 4. Cutting and drilling template for the servo mounts.

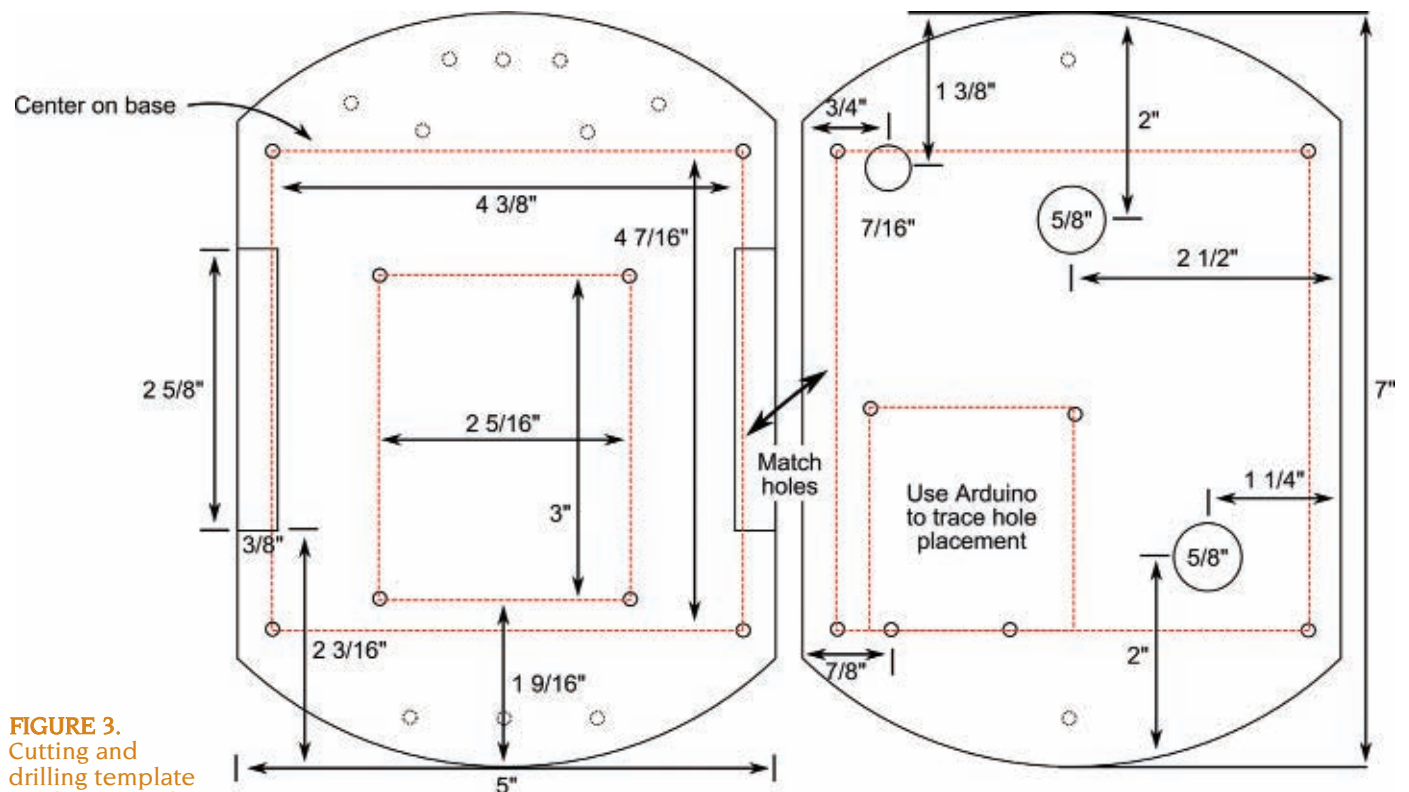
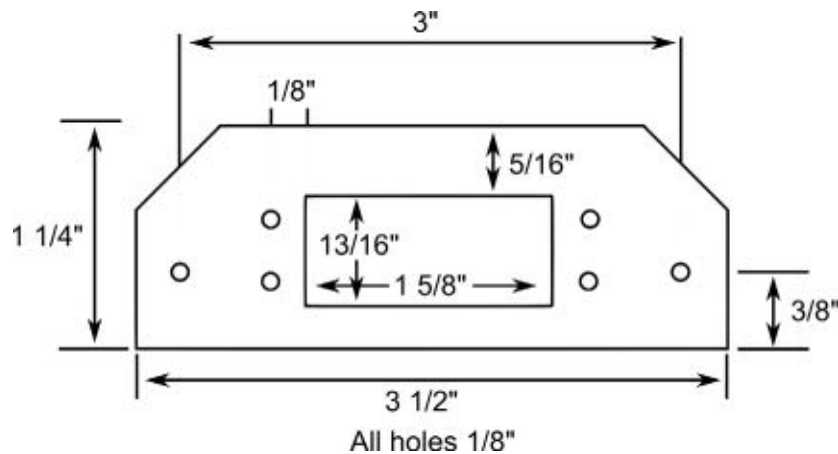


FIGURE 3. Cutting and drilling template for the top and bottom decks for the ArdBot II.

All holes 1/8" unless noted
Holes bounded in red require critical placement.
Holes in dotted outline are optional. Drill to suit.

your computer (we'll get to why you might want to do this in the next installment). By simply disconnecting the batteries for the motors, your ArdBot II will *think* it's driving around, but, in fact, won't move.

Another important reason to use separate supplies is changing out those un rechargeable 1.5 volt AA cells with rechargeable ones. In the long run, it'll save you a ton of money — robots use up batteries like kids eat candy. The rechargeables are also better on the environment.

The nominal voltage output of nickel cadmium or nickel-metal hydride cells is 1.2 volts; four together provide 4.8 volts. That's enough to motivate the robot's two servo motors, but not enough to power the Arduino's regulator.

Making Your Own ArdBot II

With the design basics out of the way, let's turn to building the ArdBot II. Construction is simple, requiring only one square foot of material. You need four pieces total: one for the bottom base, with small cut-outs for the wheels; one for the top base; and two for the servo mounts.

A cutting and drilling diagram for the main top and bottom decks is provided in **Figure 3**; **Figure 4** shows a cutting and drilling guide for the servo mounts.

You can use 1/4" plywood or my preference: 6 mm expanded rigid PVC plastic. The plastic is lighter weight, and if purchased locally from a plastics distributor, is often cheaper than wood. Either way, plywood and PVC can be readily cut using ordinary wood tools. I suggest a coping saw for working by hand, but recommend a motorized table scroll saw for the fastest and most accurate cutting.

Servos and servo mount hardware	
12	4-40 x 1/2" pan head machine screw
4	4-40 x 1/2" flat head machine screw
12	4-40 hex nut
4	3/4" x 3/4" plastic angle bracket*
Arduino mounting standoff hardware	
4	3/4" to 1" (approx.) length nylon standoffs, 4-40 thread
4	4-40 x 1/4" pan head nylon machine screw
4	4-40 x 1/2" flat head machine screw
Skids hardware	
2	6-32 cap (acorn) nut
2	6-32 x 3/4" pan head machine screw
4	6-32 hex nut
Deck risers (standoffs)	
4	Standoffs with 4-40 or 6-32 threads, at least 2" or longer
8	4-40 or 6-32 pan head machine screws
* For the 3/4" x 3/4" plastic angle brackets, you can substitute the nearest smallest metal L brackets available. The plastic brackets weigh a little less.	

Table 1 ▲

You can use any type of drill — all holes are 1/8" unless otherwise indicated. Some hole locations are critical, so follow the old adage: Measure twice, cut (or drill) once.

When marking off the holes for the Arduino in the top deck, use your Uno board for the hole locations. I include all four mounting holes, although on the latest Unos you're likely to only use three of them since the one in the upper left corner is very close to the top row of headers. The hole is still there in the board in case you want to mount it with a shield stacked on top.

If you'd prefer not to do the cutting and drilling, see the **Sources** box for a kit of parts, including pre-cut body pieces and assembly hardware.

To complete the ArdBot II, you'll need the hardware listed in **Table 1** to put all the pieces together. Except for the skids and riser standoffs, I've specified 4-40 size hardware for everything. While 4-40 machine screws and nuts are a little harder to find than their more common 6-32 siblings, their smaller size makes them better candidates for a small robot. They also weigh less, and the less weight a robot has, the longer its batteries will last.

If your local home improvement store doesn't have the 4-40 hardware, you can always get it online; see the **Sources** box for a small selection of Web retailers that carry miniature fasteners.

Tip: When searching the local digs for miniature nuts and screws, try the smaller hardware stores first. They're more likely than the "big box" retailers to have what you're looking for.

In addition to the assembly hardware, *at a minimum* you need the additional components in **Table 2** to build your ArdBot II. Some additional parts will be needed to complete other functionality of the robot. We'll cover these

2	Standard size servos, continuous rotation, Futaba compatible splined hubs
2	2-1/2" or 2-5/8" wheels with Futaba compatible hubs
1	Arduino Uno microcontroller board
1	Mini solderless breadboard (170 tie points, double-sided foam tape on back)
1	4 x AA battery holder
4	AA batteries (alkaline or rechargeable)
1	Nine volt battery snap and wires (lead length should be at least 6")
1	Nine volt battery
1	2.1 mm barrel solderable connector
1	Miniature piezo speaker
2	Snap action leaf switches (standard size)
1	12" length three-wire (signal, power, ground) extension
Misc	Pre-made female-female jumper wires* (some will be cut to length)
Misc	Set of 10 or more double-length break-away header pins
Misc	12" length of 0.25" OD (outside diameter) aquarium tubing**
* Pre-made female-female jumper wires and double-length break-away headers are available from Pololu. I recommend the 50 pack of 3" length female-female jumper wires (rainbow assortment), part number 1706. This is far more than you'll use for the ArdBot II, but the extra will always come in handy.	
** This tubing is used for extending the contact area of the leaf switches. We'll get to this aspect in an upcoming installment of the series.	

Table 2 ➤

Making Your Own Deck Risers

The ArdBot II uses risers — also called standoffs — to separate the top and bottom decks. Long (2" or more) standoffs for the deck risers can be hard to find and expensive. Full retail for each standoff may cost a dollar or more, and you need four of them. Options include screwing together shorter standoffs or use long 6-32 machine screws and nuts, along with aquarium tubing as spacers.

A 3" machine screw will provide a riser length of about 2", which is perfect. You need to allow for the thickness of the two body pieces (about 1/2" total), plus another 3/8" or so for securing the nuts. To use this method, thread the screws up from the bottom, and attach the precut lengths of tubing. Lightly compress the tubing by hand-tightening a nut over it. Do the same for all four risers. When it's time to attach the top deck, position it over the machine screws and secure with four more nuts. Hand tighten only.

parts in upcoming installments as they are described and demonstrated.

Assembling Your Robot

With the body pieces constructed (or purchased) and all other parts in hand, you're ready to build your ArdBot II. Note that not all of the components listed are used in this first article. Some (like the piezo speaker and switches) are explained in more detail in upcoming articles in the series.

Step 0

Before assembly, you may want to use 150 grit sandpaper to smooth the edges of the base parts.

Orient the bottom deck so that the holes are aligned as shown in **Figure 3**. Note that the holes for each servo are not symmetrically placed on the deck. This is to accommodate the offset of the servo drive shaft. While there is technically no "front" or "rear" of the ArdBot II, for the purposes of assembly, the top of the illustration in **Figure 3** is the front, and the bottom is the rear.

Step 1

Insert a servo into a servo mount by sliding it back-end first through the mount. The fit may be tight, depending on the make and model of the servo. (As necessary, enlarge the rectangle for the servo using a file or coarse sandpaper.) Do not force the servo into the mount, or the mount may be damaged.

Secure the servo to the mount with 4-40 x 1/2" screws and hex nuts (**Figure 5**). You can use either four screws for each servo or only two. When using two screws, position them on opposite corners of the servo mounting flange as shown.

Repeat for the opposite servo and mount. *Be sure to construct the second servo and mount in mirror-image to the first!* Refer to **Figure 7** in Step 3 to see how the motors should be inserted into the mounts.

Step 2

Using 4-40 x 1/2" machine screws and nuts, attach two plastic L brackets to each of the servo mounts. You'll make a "left" and a "right" mount assembly.

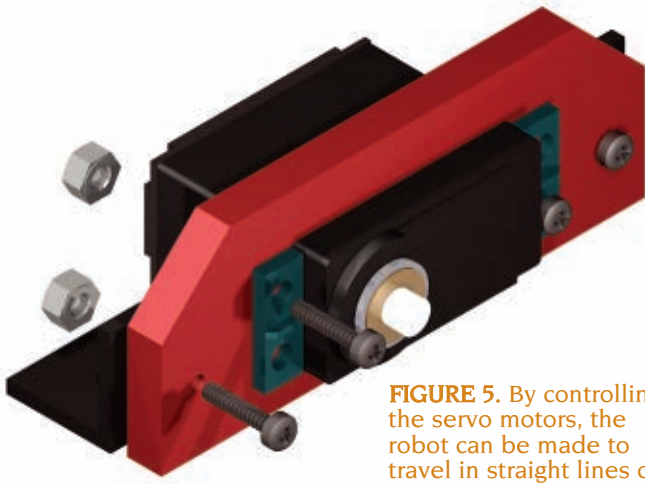


FIGURE 5. By controlling the servo motors, the robot can be made to travel in straight lines or make turns. By reversing the motors relative to one another, the robot will spin in place.

FIGURE 6. Each servo mount attaches to the bottom base of the ArdBot II with a pair of L brackets.

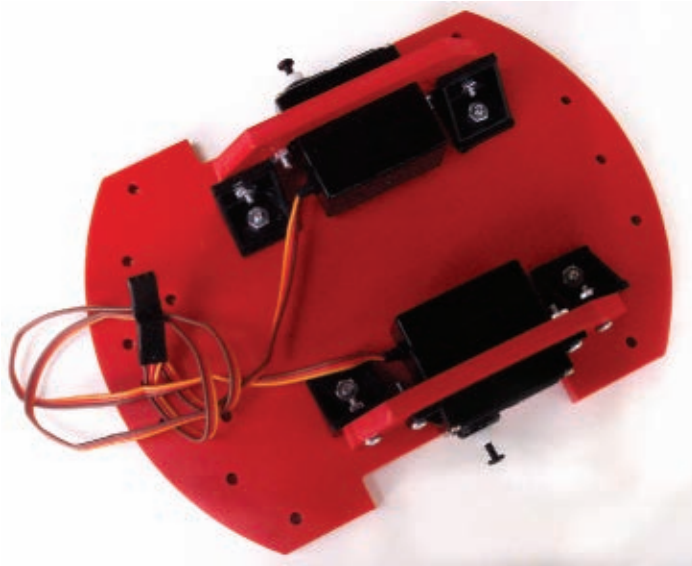
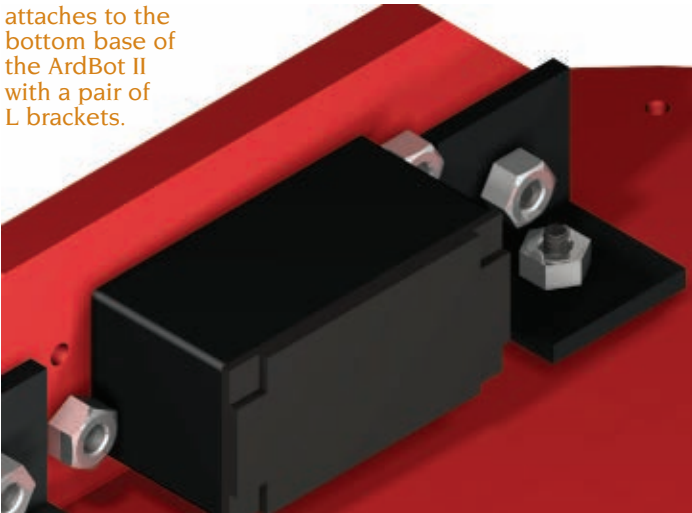


FIGURE 7. The ArdBot II bottom base showing mounted servo motors. Note the orientation of the motor shafts relative to the front of the robot (the front is pointing left).

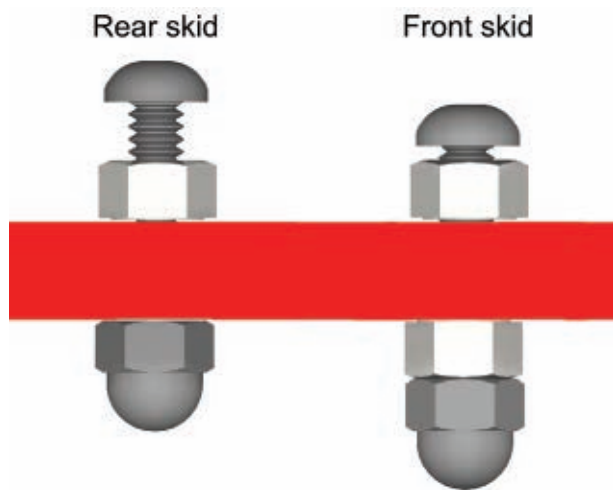


FIGURE 8. The front and back of the robot is leveled using skids. Adjust the length of the skids so that the robot will slightly tip back and forth along its length.

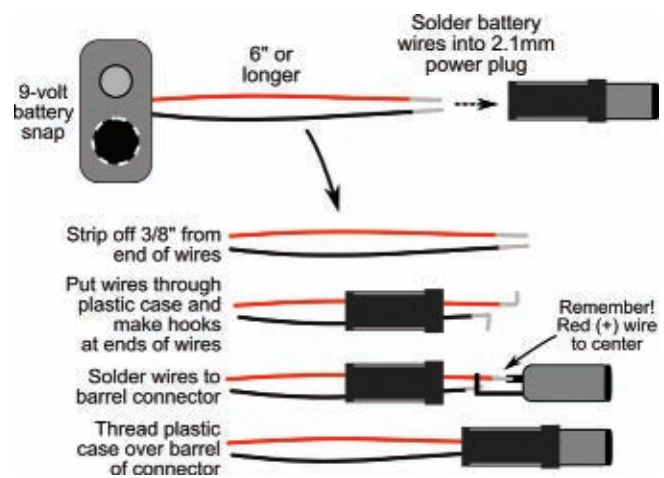


FIGURE 9. Solder a 2.1 mm barrel connector to the nine volt battery leads as shown. Be sure to double-check your work against shorts, and that the red lead is soldered to the center of the connector.

- For the left mount assembly, the motor shaft should face to the left, and toward the “top” of the deck (as referenced in **Figure 3**). Attach the L brackets to the right side of the mount.
- For the right mount assembly, the motor shaft should face to the right, also toward the top of the deck. Attach the L brackets to the left side of the mount.

Insert the machine screws through the L bracket, then through the servo mount. Secure on the other end with a nut. Before tightening, be sure the bottom of the L bracket is flush with the bottom edge of the servo mount.

Step 3

Attach the left mount assembly to the bottom deck using 4-40 x 1/2” screws and nuts. The screws should come up from the underside of the deck, through the L bracket as shown in **Figure 6**. When orienting the mount assembly, be sure that the servo shaft is centered in the wheel well cut-out. Align the assembly so it is parallel with the wheel well cut-out, then tighten all screws.

Repeat the same procedure for the right mount assembly. **Figure 7** shows how the completed servos and mounts should look on the bottom deck of the robot.

Step 4

Attach the front and rear skids as shown in **Figure 8**. Each skid uses a 6-32” machine screw, hex nut, and acorn (cap) nut.

1. Thread a machine screw into the hole at the front and back base. The screw is inserted from the top of the deck (the side with the servos).

2. Put the hex nut onto the screw, followed by the acorn nut. Tighten the acorn nut against the hex nut.

Repeat these steps for the other skid.

You can adjust the height of the skid by loosening or tightening the machine screw in the hole. If you need

greater height adjustment or the hole for the skid is too large to self-tap, merely use a longer machine screw, and tighten into place using nuts on both the top and bottom of the deck.

Important: Properly level the robot so there is only a slight “teeter tot” forward and back. Otherwise, the bot may not steer properly, and it could get caught up in even the smallest of obstacles. If there’s too much teeter tot movement, adjust the skids so they are closer in length to one another. If there’s not enough movement, lengthen one skid.

Step 5

Attach the wheels to the servos. Each wheel is secured with a small self-tapping screw that is supplied with the servo. Note that the servo shaft is splined, and this spline matches the wheel hub. Be sure to press the wheel onto



FIGURE 10. Mount the nine volt battery to the bottom base using a small piece of double-sided foam tape.

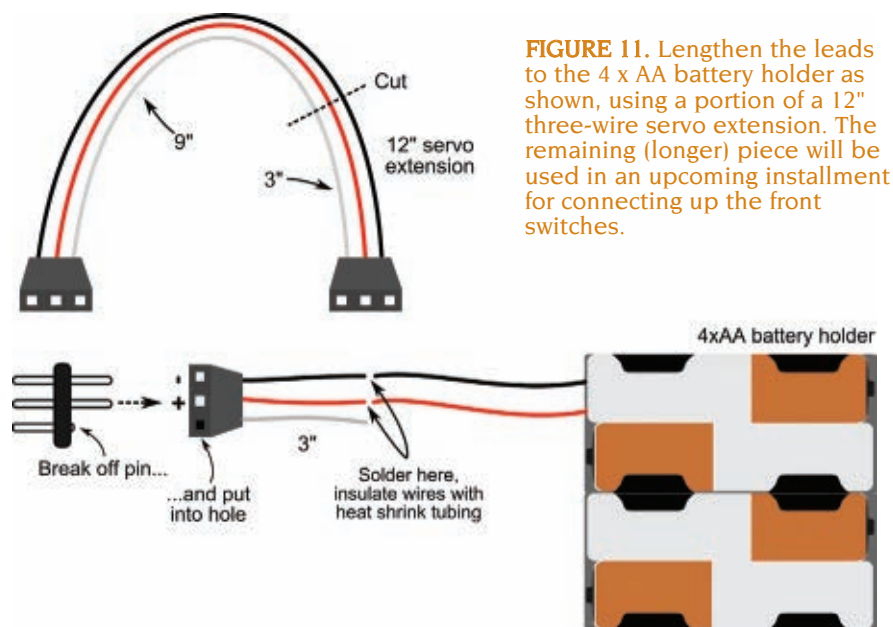


FIGURE 11. Lengthen the leads to the 4 x AA battery holder as shown, using a portion of a 12" three-wire servo extension. The remaining (longer) piece will be used in an upcoming installment for connecting up the front switches.

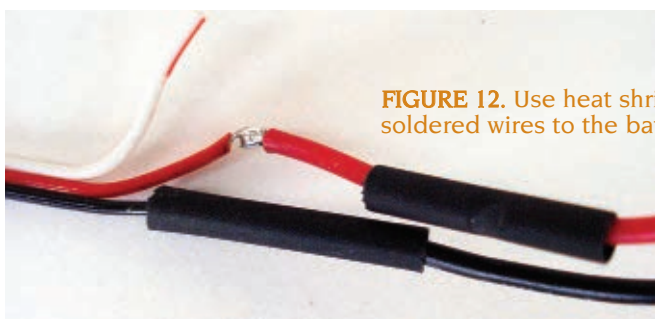


FIGURE 12. Use heat shrink tubing on the soldered wires to the battery holder.

the shaft firmly while tightening the screw. Do not over-tighten the wheel mounting screw, but be sure the wheel is on snugly.



FIGURE 13. Mount the 4 x AA battery holder over the center of the robot using Velcro or other hook-and-loop fastener. The battery holder will sit atop the servo motors.

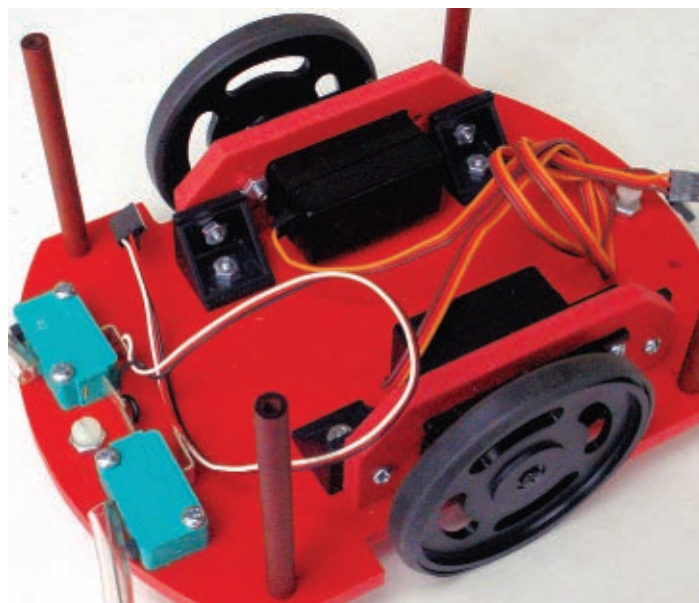


FIGURE 14. The completed bottom deck of the ArdBot II, also showing the twin leaf switches. These will be detailed in an upcoming installment.

Sources

Budget Robotics

Precut ArdBot II body chassis, with all assembly hardware
www.budgetrobotics.com

Jameco

Continuous rotation servos, servo wheels, piezo speaker, Arduino, mini breadboard, miniature hardware.
www.jameco.com

Micro Fasteners

Large variety of miniature (4-40 and smaller) nuts, screws, and other hardware.
www.microfasteners.com

Parallax

Continuous rotation servos, servo wheels, piezo speaker, three-wire servo extension.
www.parallax.com

Pololu

Continuous rotation servos, servo wheels, piezo speaker, Arduino, mini breadboard, miniature hardware.
www.pololu.com

Step 6

Follow the steps in **Figure 9** to attach a 2.1 mm barrel connector to the end of a nine volt battery clip and wires. Remember that the red (positive) lead must be soldered to the center conductor of the connector. For best results, the leads on the battery clip should be at least six inches; seven to nine inches is even better.

Tip: Use your volt-ohm meter to test for shorts. Dial it to continuity, and without a battery attached on the other

end, test the center and outer sleeve of the connector. The reading should show infinite ohms.

Step 7

Secure the nine volt battery clip to the nine volt battery. Use a small piece of double-sided foam tape to mount the battery to the center of the bottom deck. The tape should not be larger than about 1/4" square, so you can readily peel the battery off when you need to replace it. **Figure 10** shows the mounting position for the battery on the robot's lower deck, between the two servo mounts.

Step 8

Follow the diagram in **Figure 11** to complete the wiring of the four AA battery holder. Start by cutting off 3" from a 12" three-wire extension set (typically used to extend the length of servo cables). Clip off the white wire, and solder the red and black wires together as shown in **Figure 12**. Apply heat shrink tubing to the soldered joints to avoid shorts (refer again to **Figure 12**). Secure the holder to the robot using small strips of Velcro™ or other hook-and-loop fastener. Apply the strips to the tops of the servos as depicted in **Figure 13**, and mount the holder over the nine volt battery. Insert fresh batteries into the four-cell holder.

Figure 14 shows the completed bottom deck of the ArdBot II with motors, mounts, and wheels attached. Note: I'm also showing the ArdBot II with the two front switches already attached. You'll do this part the next time.

Step 9

Mark off the holes for the Arduino in the location shown back in **Figure 3**. Drill the three holes using a 9/64" bit. Secure the Arduino board to the top deck using 4-40 screws and standoffs. **Important!** If you use metal screws to hold the Arduino, be sure to add plastic washers to avoid any possible shorts. For convenience sake, you may wish to opt for nylon screws. Mount the mini solderless breadboard next to the Arduino. Most mini breadboards come with double-sided self-adhesive tape. If yours doesn't, you'll need to add your own.

Step 10

To complete the ArdBot, secure the top deck to the riser standoffs using 4-40 or 6/32" x 1/2" screws (or review the **sidebar** on how to use long screws to make your own standoffs).

Coming Up: Finishing Construction and First Trial Run

That's all the space we have for this first installment. Next time, we'll complete the electrical wiring, show how to attach the batteries and servos to the ArdBot, and how to do a simple test to make sure everything is working as it should. We'll then move on to programming the Arduino to detect obstacles, control the robot remotely, sense when it's been picked up or moved, and lots more. **SV**

STEER WINNING ROBOTS WITHOUT SERVOS!



Perform proportional speed, direction, and steering with only two Radio/Control channels for vehicles using two separate brush-type electric motors mounted right and left with our **mixing RDFR dual speed control**. Used in many successful competitive robots. Single joystick operation: up goes straight ahead, down is reverse. Pure right or left twirls vehicle as motors turn opposite directions. In between stick positions completely proportional. Plugs in like a servo to your Futaba, JR, Hitec, or similar radio. Compatible with gyro steering stabilization. Various volt and amp sizes available. The RDFR47E 55V 75A per motor unit pictured above.

www.vantec.com

VANTEC Order at
(800) 882-6832

DIY NEVER HAD A BETTER FRIEND...

Take your DIY project to the next level with PanaVise work holding tools. Perfect for hobbies, soldering, electronics, crafting & repair, PanaVise tools have hundreds of uses only limited by your imagination.

PanaVise, "better than a third hand!"



PANAVISE
Innovative Holding Solutions

Now available at



Model 301
640-0178



Model 324
640-0177



www.radioshack.com/StoreLocator

It's All in Your Head:

How Scientists are Looking to the Anatomy of the Human Body to Make Smarter and Faster Machines

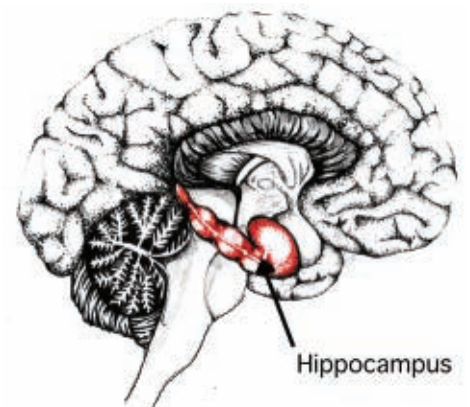
by Morgan Berry

As an undergraduate psychology student, I am currently struggling through one of the tougher major requirements: the dreaded Physiological Psychology course. Essentially an anatomy course focused exclusively on the nervous system, Physio Psych (as we call it for short) delves into the minutest of the minutia of the human body.

My study sessions are spent trying to commit to memory the difference between the inferior colliculus and the superior colliculus, the precentral gyrus and the postcentral gyrus, and the tectum and the tegmentum — apparently, name variation was not on the agenda when scientists were naming the parts of the body.

As I stare at diagrams of the brain, the one question that keeps plaguing me is, "Why on earth do the body parts look like they do?" For example, why does the cerebellum — Latin for "little brain" — look so much like, well, a little brain? Why does the hippocampus look so much like a seahorse?

For every single body part, the texture, size, shape, color, and orientation evolved over millions of years for a specific purpose, and the variety that emerged is simply astounding. It is no wonder that today's leading artificial intelligence scientists are taking a cue from the body as well, aiming to mimic the natural systems humans possess in order to enhance their capabilities.



The hippocampus gets its name because of its shape, which some people think looks like a seahorse (in Greek mythology, Hippocampus was the name of a monster-like half horse, half fish). See the resemblance?



The field of embodied cognition aims to do just that. The theory these philosophers and scientists work from is that the mind and body are inextricably linked. The structure of our body heavily influences our thought processes.

Hearing this, I'm reminded of the message my high school biology teacher drilled into our heads almost every lesson: Structure is related to function. If someone wants to understand how the brain works, then it is crucial to look at how it is designed. Scientists that study embodied cognition stress that the brain alone is not responsible for our behavior. The brain is just one portion of a greater system, including the nervous system and the environment that produces opportunities to behave a certain way.

So, how does this play into many scientist's desire to create a walking, talking, intelligent humanoid robot?

Biorobotics

Biorobotics — an interdisciplinary field that aims to mimic biology when creating robots — relies on the theory of embodied cognition as a solution

to the real world problems that come with an artificial machine trying to operate in a very human world.

Traditionally, a robot attempting to mimic human behavior would be designed mainly using software to try and simulate the human brain. When confronted with a situation, the robot turns to its base of knowledge to try and solve it. For example, when put in front of a staircase, the robot would turn to its programming to recall how it is supposed to climb stairs, where it is supposed to position its feet, and how to coordinate the movement of its legs to successfully make it up the stairs.

The problem with this method is that it ignores the fact that the brain is just one part of the process. The sensory neurons in the feet tell the body where it is on the stairs; the inner ear helps maintain balance; and reflexes — which are actually neural networks that operate independently of the brain's input — step in to rescue the body if something goes wrong.

This model of artificial intelligence in robotics design has proven to be somewhat inadequate. For instance, there are numerous cringe-worthy videos on YouTube of the Honda

Asimo robot falling down the stairs during a demonstration. Contrast this with a robot more in line with the embodied cognition approach. The BigDog — designed by Boston Dynamics for military use — attempted to mimic the structure of a mammalian body in order to adapt to uncertain terrains.

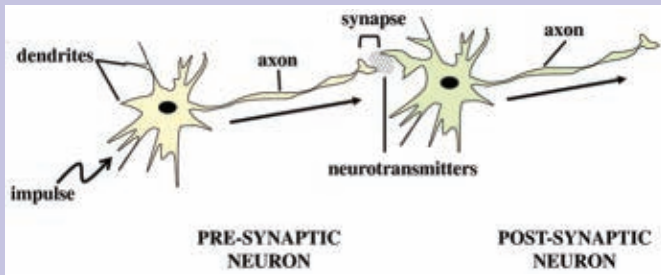
According to Boston Dynamics, the legs of the BigDog are designed to mimic an animal's, with shock absorbing joints and sensors to monitor the position of the legs, the amount of energy being used, and the current force on the joints. Watch a YouTube clip of the BigDog in action, and you'll see an eerily animal-like walking pattern. Kick the BigDog and it will catch itself without tipping over. Put it on an icy path and it will stumble and slip the way humans do. (It manages not to lose its balance and keeps on walking.)

The Cheetah — another robot designed by Boston Dynamics that mimics the natural movements of the animal — can run 28.3 mph which is a little faster than Usain Bolt, the company boasts. Apparently, there is a lot more to artificial intelligence than cognition.

Boston Dynamic's BigDog is one of the all-terrain robots that take their cue from our anatomy.
Photo courtesy of Boston Dynamics.



Neural Networks 101



This is an artist's interpretation of two neurons communicating. Note: These parts are not drawn to scale. In the body, the axon can be one meter or longer.

Assuming you aren't a current psychology or medical student, you may be a little fuzzy on the way our body transmits messages through the nervous system. So, here is a crash course on how the process works — just in case high school biology was a little further back in the past than you care to admit:

- Neurons are the cells in your body that help the brain communicate with the muscles, and vice versa. The cells communicate with each other using chemicals (called neurotransmitters) that are transmitted from neuron to neuron.
- The main parts of a neuron are the cell body, the axon, the dendrites, and the end bulbs (refer to the **figure**). Basically, a neuron receives a message through the dendrites, sends it down the axon, and sends it to the next neuron through its end bulbs.

- Neural networks work by sending chemicals across the gap between the end bulbs of one neuron and the dendrites of the other, called a synapse. Once the message has been sent across the synapse, the journey ends for that batch of neurotransmitters. After the message has been communicated down the next neuron, that neuron makes more chemicals for the next neuron, and so on.

Picture a line of dominoes. The first domino causes the next one to fall, then the next, all down the line. This system differs from an electrical one, where the same signal is transmitted all the way down the line. It is crucial that the signal is "recreated" at each step of the process. Otherwise, a signal sent from a neuron in your shoulder to the brain would be stronger than one sent from your toes. In other words, your shoulder would be more sensitive to touch than your toes, which is not how we want our bodies to work!

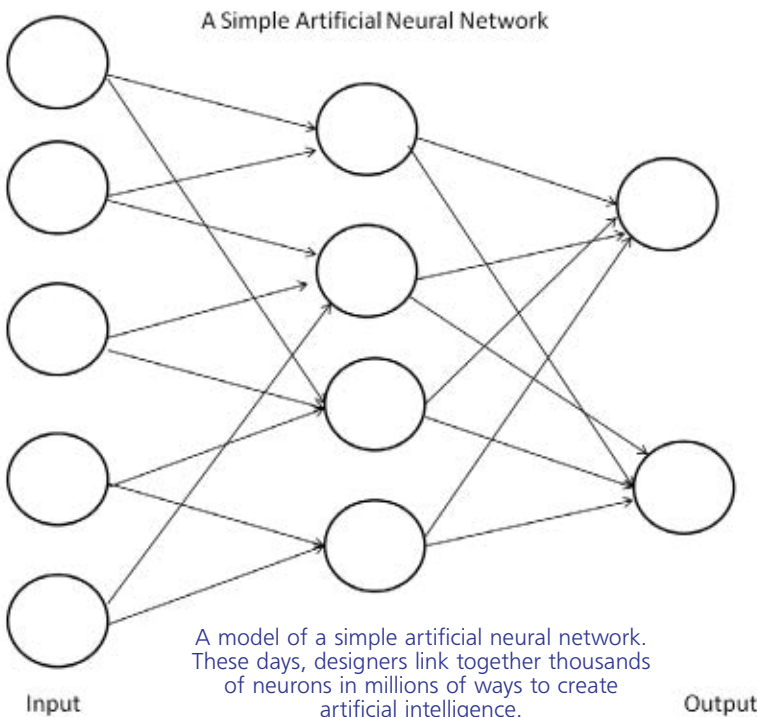
- Neurons require a certain amount of stimulation in order to fire to pass their threshold. Say, a neuron needs a signal of at least +2 to fire (this is completely arbitrary). If it only gets a signal of +1, the message won't be sent to the next neuron. However, the effects of neurons can accumulate. If a neuron is getting a signal of +1 from three different neurons, then the effects will add up and the neuron will send the message. Similarly, if a single neuron keeps sending a signal of +1 over and over again, the signals will add up to go past the threshold.

Here's the thing: Sometimes neurons send excitatory signals (ones that tell the neuron to fire) and sometimes they send inhibitory signals (ones that say *not* to fire). These inhibitory and excitatory signals can add up in countless ways, thus creating a huge variety in the way our body communicates with itself.

ANN

One of the more esoteric ways human anatomy has inspired artificial intelligence work is the Artificial Neural Network (ANN). This is not a neural network in the literal

sense, so don't go thinking that scientists are developing robots that can feel, smell, and taste (although a robot has been developed in Japan that can simulate feeling pain in order to train dental students to be more gentle when dealing with patients, but this isn't the same thing).



An ANN is actually a mathematical model that takes its inspiration from the human nervous system (for a refresher course on the anatomy of the nervous system, check out "Neural Networks 101"). ANNs work much in the same way that actual neural networks do. An ANN has neurons that communicate messages with each other (see **graphic**). Like a real neural network, some of these are excitatory and some are inhibitory. Some are weighted as being more important than the others.

When a neuron gets a signal from the neurons around it, it adds up the signals and compares it to a threshold which determines whether or not it will fire. It's actually a pretty simple model on the individual level.

The real power comes from the networks as a whole. Through this pattern of weighting some neurons as positive, some as negative, and some as neutral, a vast number of messages can be sent through the network. Because the individual neurons are constantly comparing the input they receive to the output they send, the networks actually begin to evolve, adapting to their particular circumstance.

This is not simple in the least, and involves an incredibly complex pattern. The idea has been around since 1943, and has, of course, gained a lot of complexity with the advent of computers and complex algorithms that can put these networks into place.

ANNs have endless real world applications, from predicting stock changes to targeting a particular demographic in advertising, but our interest here is how they can be applied to artificial intelligence in robotics. One use is as a model for automated chatterbots. For example, a chatterbot placed online to guide users through tutorials during off hours could benefit from the adaptability and realism of an ANN.

Eventually, this type of programming may translate into other types of robots too, especially human-like ones built primarily for communication. Combined with the principles of embodied cognition, the next generation of robots will likely be smarter, faster, and more efficient than we could have imagined.

DNA

Besides AI, there is much interest in adapting other bodily systems for human purposes, as well. Scientists made headlines in January when they announced that they had managed to store information — Shakespeare's 154 sonnets, a scientific paper, a clip of the "I Have a Dream Speech," and a color photograph, to be exact — on a tiny portion of synthetic DNA. The scientists then read the information back with 100% accuracy.

Although scientists had previously encoded text into DNA, this attempt was unique for the fact that the scientists also encoded a photograph and audio information.

The process works much in the same way that biological DNA does: A machine encodes the information into the same series of A, T, C, and G that the body uses to store the genome. The machine works much like an inkjet printer, but rather than using ink, the very same chemicals that make up the nucleotides in our body

are encoded and structured to make the synthetic DNA.

DNA storage has exciting implications. The researchers estimate that everything humans have ever written could be encoded to DNA and weigh less than a granola bar. Perhaps even more important is how stable DNA storage is. In a cool, dark, dry room, DNA could preserve human knowledge for thousands of years.

Currently, the high cost of the encoding process makes this storage method prohibitive, but in a decade or so, people could quite possibly store their photographs and school work on a tiny speck of DNA, leaving USB drives in the dust.

Final Examination

More and more, scientists are turning to nature to discover more efficient ways to design machines. Considering that the human body has been "fine-tuned" over millions of years of evolution to its current state, it makes sense that the human body is now being seen as a model of efficiency to be mimicked and altered to our needs. Just look at the complexity of the human nervous system, for example, and you can see how amazing our body really is.

Messages travel through our body nearly instantaneously — something scientists aren't even close to being able to replicate. Humans are by no means perfect, but when you sit back and admire how amazing it is that we do what we do, then it becomes clear that the road to scientific exploration also begins with studying and understanding the systems within our bodies.

With this knowledge in mind, I'll return to my *Physiological Psychology* book with a new appreciation for the material I'm learning, because who knows ... maybe the next scientific breakthrough is somewhere right on those pages. **SV**

Firgelli
www.firgelli.com

LINEAR SERVOS



L12-R Linear Servo

- Direct replacement for regular rotary servos
- Standard 3 wire connectors
- Compatible with most R/C receivers
- 1-2ms PWM control signal, 6v power
- 1", 2" and 4" strokes
- 3-10 lbs. force range
- 1/4" to 1" per second speed ranges
- Compatible with VEX



L16 Linear Actuators

- 2", 4" and 6" strokes
- 10 - 40 lbs. force range
- 1/2" to 1" per second speed ranges
- Options include Limit Switches and Position Feedback

New!

PQ12 Linear Actuator

- Miniature Linear Motion Devices
- 6 or 12 volts, 3/4" stroke
- Up to 5 lbs. force
- Integrated position feedback or limit switches at end of stroke
- External position control available



Linear Actuator Controller (LAC)

- Will drive any Linear Actuator with position feedback
- Up to 24v and 4 Amps
- USB connectivity to drive the actuator with your computer
- Adjustable speed, limits and sensitivity



L12-NXT Linear Servo

- Designed for LEGO Mindstorms NXT®
- Plugs directly into your NXT Brick
- NXT-G Block available for download
- Can be used with Technic and PF
- Max. speed: 1/2" per sec.
- Pushes up to 5 lbs.
- 2" and 4" strokes



Available Now @
www.firgelli.com

Tibbo Trouble

I just left a friend of mine that is a master machinist. I told him that I was on the way back to my shop to finish up a SERVO article. He asked what it was about and what programming language I would be using. When I told him Basic, he replied, "Good ... That's something I understand." The funny thing about my friend is that he has absolutely no problem building complicated machined parts that must hold tolerances of ± 0.0001 of an inch, and he regularly uses the Basic language to write scripts for his custom-made embedded CNC machine tools. That in itself says a lot for the Basic language.

by Fred Eady

You can discuss this topic at <http://forum.servomagazine.com>.

We are going to examine a development system that feels and smells like Basic, but is capable of performing C-like tricks. The combined hardware and Basic development environment falls under the banner of Tibbo.

Tibbo Basic 101

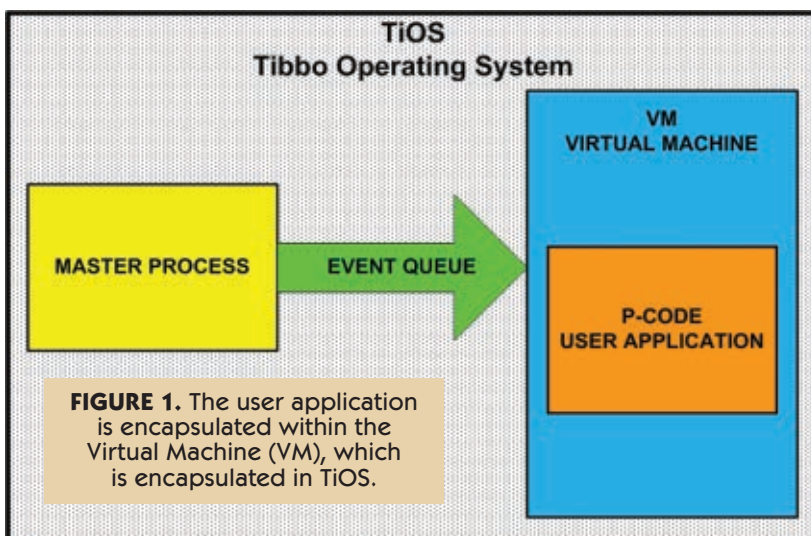
Figure 1 is a graphical representation of the Tibbo firmware environment that runs on the target Tibbo hardware. The Master and VM (Virtual Machine) processes run under the control of TiOs (Tibbo Operating System). From a process point of view, the Master Process is in charge of communications and is the link to TIDE (Tibbo Integrated Development Environment).

Events are also generated by the Master Process. VM — which executes the compiled user application — runs under the control of the Master Process.

The V in VM means that the execution engine for the user application contains no silicon. VM is chunks of firmware and hardware that emulate a silicon-based hardware processor. The user application is compiled into Tibbo P-Code using the compiler component of TIDE. P-Code is just another name for pseudo-code. P-Code is not machine code and cannot be executed by a piece of hardware.

VM interprets the P-Code in a way that emulates the execution of machine language code. There are many advantages to running the Tibbo user application under VM. If the user application bombs, it will only affect the VM process at the most. The Master Process and TiOs will continue to run, isolating the target hardware from the application crash. Since the Master Process has total control of the Virtual Machine, the Master Process can stop the VM and send debug information to the programmer via TIDE.

Tibbo user applications are event-driven. Events such as incoming characters, logic level transitions, and timer overflows are detected by the Master Process. These events are passed to



the VM by way of the Event Queue. VM pulls the events from the Event Queue and passes them on to the user application for handling.

The Flavor of Tibbo Basic GPIO

Tibbo Basic functionality is based on platforms. For instance, smaller platforms may have network support but provide no support for SPI or I²C communications. All Tibbo Basic platforms support a mix of objects, variable types, and functions (syscalls). Each Basic platform is also backed by platform-specific constants.

Most of our Basic programming will be done against objects. Reading and writing the I/O pins falls under the *io* object. Suppose we want to control the state of GPIO 2. There are many ways to do this, depending on our application needs. We can choose to manipulate the I/O port with pre-selected object methods or object methods that do not require pre-selection. Here's an example of pre-selection port manipulation:

```
io.num = PL_IO_NUM_2
io.state = HIGH
io.state = LOW
```

In our pre-selection code, we must first specify the GPIO pin number using the *io.num* method. The *io.num* GPIO pin code description translates like this:

```
Selected I/O Pin Number = Platform I/O Pin
Number 2
```

Once *io.num* is used to select a GPIO pin, the pin selection will not change until the I/O pin argument is changed with another *io.num* call. Now, with that, each of the following *io.state* calls will work against GPIO pin 2.

Now, let's manipulate GPIO pin 2 without pre-selecting the I/O pin:

```
io.lineset(PL_IO_NUM_2,LOW)
```

We just forced GPIO pin 2 to a logically low state. We can also do that like this:

```
io.lineset(PL_IO_NUM_2,0)
```

Or, like this:

```
io.lineset(2,0)
```

A Byte of Tibbo Basic Serial

One of Tibbo Basic's strengths is that the smaller platforms can be easily configured as serial-to-Ethernet devices. The serial configuration code almost talks to you:

```
ser.num=0          'serial port number
ser.baudrate=ser.div9600 '9600 baud
```

```
ser.rxbufirq(2)    '256 bytes for
                   'receive buffer
ser.txbufirq(2)    '256 bytes for
                   'transmit buffer
ser.flowcontrol=DISABLED 'no flow control
ser.mode=PL_SER_MODE_UART 'UART Mode
ser.parity=PL_SER_PR_NONE 'no parity
ser.bits=PL_SER_BB_8      '8 data bits
sys.bufalloc       'allocate the
                   'buffers
ser.enabled=YES      'enable serial
                   'port 0
```

Accessing the serial data is also a no-brainer:

```
sub on_ser_data_arrival()
    ser.setdata(ser.getdata(255))
    ser.send
end sub
```

When incoming serial data hits the buffer, the *on_ser_data_arrival* event is fired. The *getdata* method is used to read the buffer without blocking the application execution. The transmit buffer is loaded via the serial *setdata* method and sent with the *send* method. This is a very simplistic serial echo example. Tibbo Basic has the ability to check the transmit buffer to determine if there is space to queue the data for transmission.

Baud rates can be easily changed as they are based on the baud rate divisor for 9600 baud. For example, let's change the baud rate to 57600. That works out to be six times faster than 9600 baud. So, our baud rate method would look like this:

```
ser.baudrate=ser.div9600 / 6      '57600 baud
```

That's fine for faster baud rates, but how about slower ones? Let's set up 2400 baud. On earth, 9600/2400 is equal to four. Thus, 2400 baud is four times slower than 9600 baud. So, our 2400 baud setting is:

```
ser.baudrate=ser.div9600 * 4      '2400 baud
```

Oas Tibbo Basic Blinking Lights

There are times when LEDs are more effective than displays. Tibbo Basic provides a pattern (Pat) object that gives the Basic programmer control of the common pair of LEDs and the optional seven pairs of LEDs that may be present. The LED pattern logic is determined by a string of characters and symbols contained within the *Pat.Play* method argument. Here's the key:

```
'-' = Both LEDs off
'R' = Red LED on
'G' = Green LED on
'B' = Both LEDs on
'~' = Loop the pattern
'*' = Double-speed pattern
```

There is also a provision to allow the current pattern to be interrupted by another pattern:

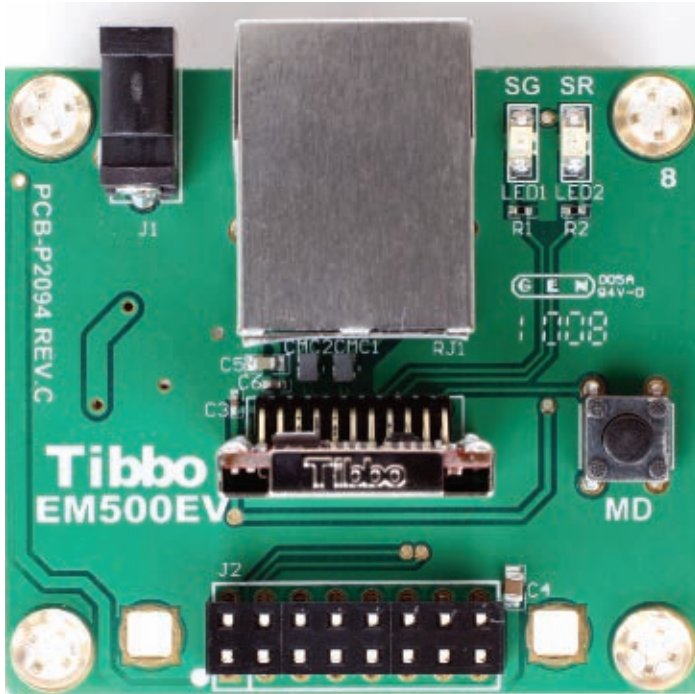


PHOTO 1. The EM500EV is basically a power supply/carrier for the EM500 MiniMo Basic-programmable Ethernet module. A switching power supply is assembled on the opposite side of the printed circuit board.

- 0 - PL_PAT_NOINT cannot interrupt current pattern
- 1 - PL_PAT_CANINT can interrupt current pattern

If we wished to blink the green LED five times at startup, the code would be as follows:

```
sub on_sys_init()
    pat.play("G-G-G-G-G-", 1)
end sub
```

Once we play the startup LED signal, let's alternately blink the red and green LEDs during normal program execution:

```
sub on_sys_init()
    pat.play("G-G-G-G-G-", 1)
end sub

sub on_pat()
    pat.play("~-G-R", 1)
end sub
```

When the initial blinking green LED pattern is finished, the *on_pat* event is fired. Note that the green/red LED pattern is looped. At this point, we can insert *pat.play* methods to indicate button events or signal incoming and outgoing data on the Ethernet or serial interfaces.

Coding Tibbo Basic Routines

The Tibbo EM500EV under the lights in **Photo 1** is based on the EM500 "MiniMo" Basic-programmable

Ethernet module which stands between the I/O header and the Ethernet magnetics. In addition to its serial and Ethernet interfaces, the MiniMo Ethernet module presents eight bits of GPIO on the EM500EV I/O connector.

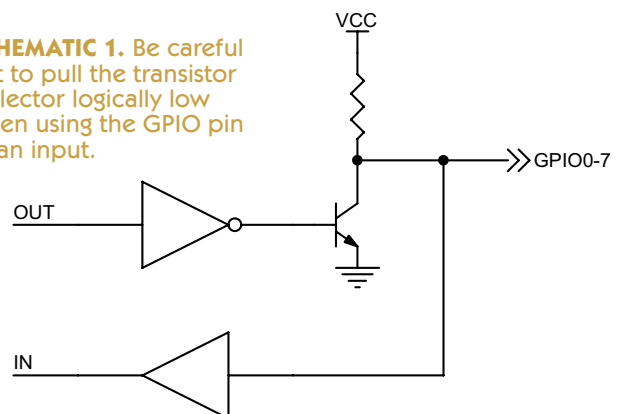
Each of the MiniMo module's I/O pins is logically represented in **Schematic 1**. The transistor and resistor make up a logic inverter. That is, when a logical high signal is presented to the base of the transistor, the transistor conducts. The conducting transistor provides a path to ground for the resistor, which pulls the transistor collector to a logically low level.

Conversely, applying a logical low to the base of the transistor turns the transistor off. With no path to ground, the transistor's collector remains at a logically high level. So, if we apply a logical high to the input of the inverter, a logical low will be presented to the transistor's base. To turn the transistor on, we must present a logical low to the inverter's input. With the addition of the input inverter, the input logic level of the inverter follows the logic level presented to the I/O connector by the transistor collector/resistor combination. The resistor also allows an external device to pull the transistor collector logically low. If we want to use the I/O pin as an input, we must first turn the transistor off (logical high applied to the inverter input). Driving the transistor collector logically high allows us to read the true state of the external logic signal.

It doesn't take much to manipulate the MiniMo Ethernet module GPIO pins. **Screenshot 1** contains code to emit a square wave from GPIO 0. The EM500EV is one of the many Tibbo Basic platforms. Each platform supports differing network interfaces, objects, and memory sizes. For instance, the EM500EV platform only supports eight bits of GPIO, 17K of RAM, 328K of Flash, and 200 bytes of EEPROM. The GPIO pins are enumerated in PL_IO_NUM:

- 0 - PL_IO_NUM_0_INT0
- 1 - PL_IO_NUM_1_INT1
- 2 - PL_IO_NUM_2
- 3 - PL_IO_NUM_3
- 4 - PL_IO_NUM_4
- 5 - PL_IO_NUM_5
- 6 - PL_IO_NUM_6
- 7 - PL_IO_NUM_7
- 8 - PL_IO_NULL

SCHEMATIC 1. Be careful not to pull the transistor collector logically low when using the GPIO pin as an input.



The politically correct way to specify GPIO pin 0 is by its proper name of `PL_IO_NUM_0_INT0`. The `INT0` in the enum name tells us that this pin can also be used as an interrupt pin. If you're planning on driving an LED with GPIO pin 0, you can hop over to the *global* tab and enter your choice of these:

```
#define led0 0
#define led0 PL_IO_NUM_0_INT0
```

As you can see in **Screenshot 1**, I chose not to use the politically correct name in my LED alias. I also chose to go without the alias in the **Screenshot 1** example.

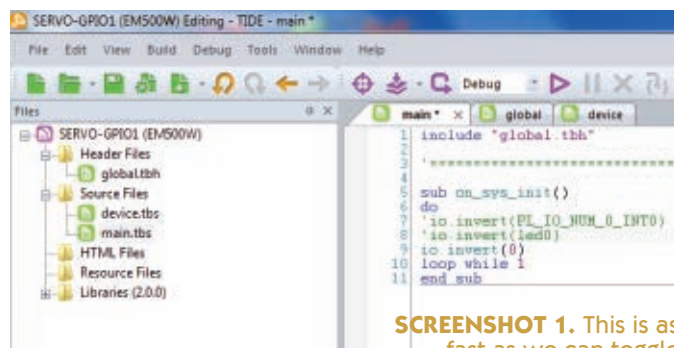
By the way, Tibbo Basic statements are not case sensitive. The bottom line is that each `io.invert` object function performs the same task. I used an eight-input Saleae Logic device to capture the pin 0 waveform you see in **Screenshot 2**. The pulses are 0.121 mS wide.

Networking with Tibbo Basic

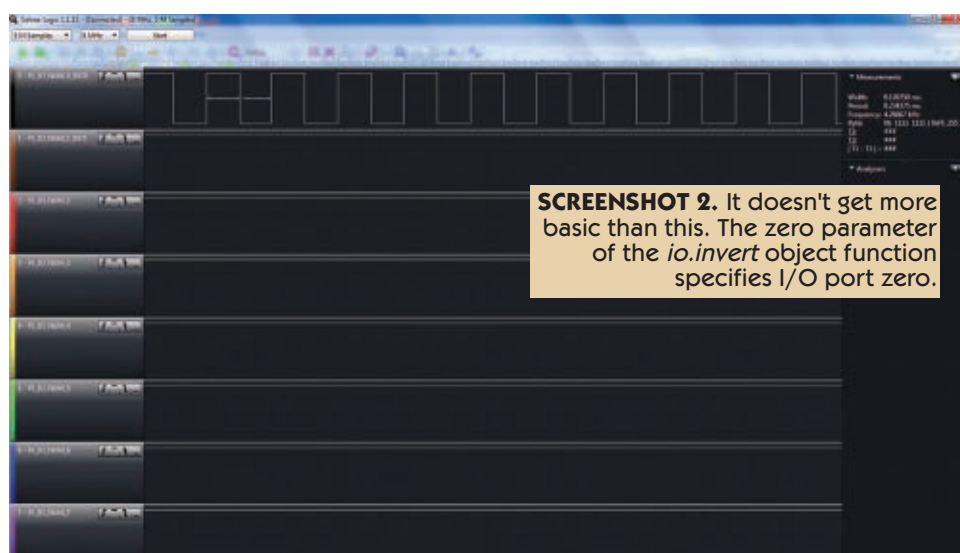
We've walked around Tibbo Basic for quite a while now. It's time to fly this thing. **Screenshot 3** is the first step towards putting together a coherent networking application with Tibbo Basic. As you can see, we've specified out a working platform (EM500W) and established a project.

In **Screenshot 4**, the Device Explorer has sensed our EM500EV and will fill in those transport blanks in **Screenshot 3**. The transport information is used to allow us to debug and program the MiniMo module using an LAN connection. Tibbo Basic uses sockets to establish TCP/IP connectivity. A socket is simply an IP address that is matched up with a port address. The combined shots that make up **Screenshot 5** demonstrate how to add socket support to our application using Tibbo Basic's `sock` library.

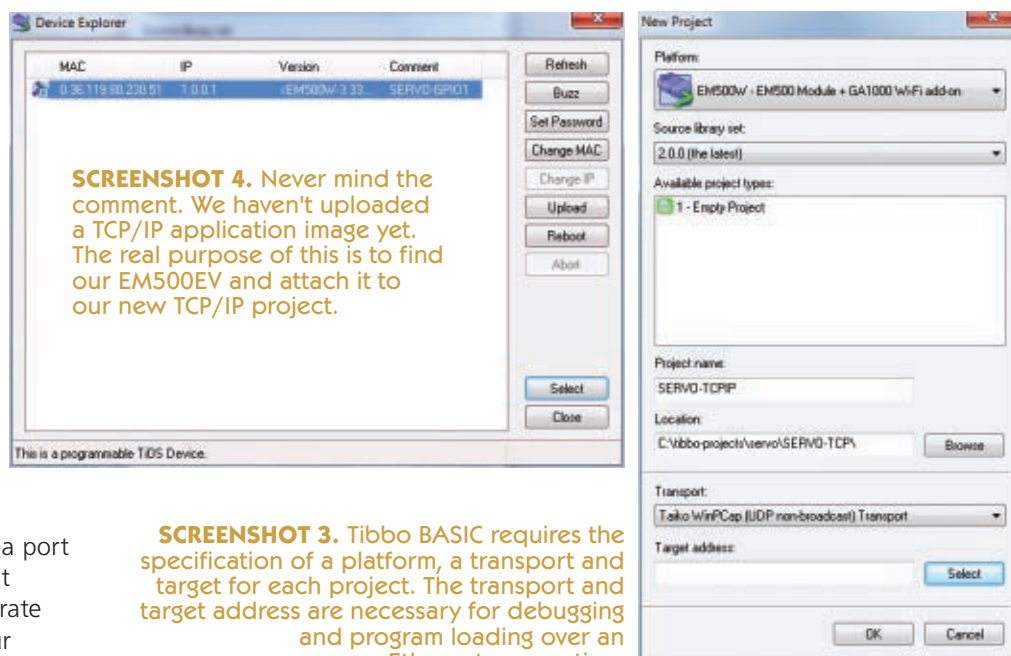
I've provided **Screenshot 6** to give you an idea of how the Tibbo Basic environment looks from a PC screen point of view. As you can see, the socket support code was



SCREENSHOT 1. This is as fast as we can toggle an EM500EV I/O pin.



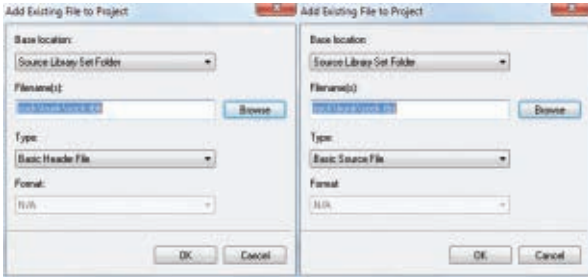
SCREENSHOT 2. It doesn't get more basic than this. The zero parameter of the `io.invert` object function specifies I/O port zero.



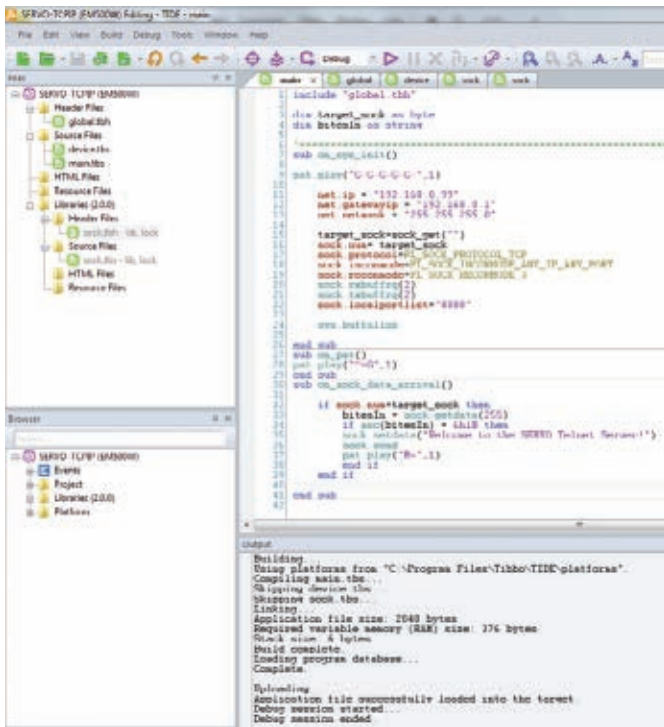
SCREENSHOT 4. Never mind the comment. We haven't uploaded a TCP/IP application image yet. The real purpose of this is to find our EM500EV and attach it to our new TCP/IP project.

SCREENSHOT 3. Tibbo BASIC requires the specification of a platform, a transport and target for each project. The transport and target address are necessary for debugging and program loading over an Ethernet connection.

added to TIDE via the work we performed in **Screenshot 5**. The only relevant user code you don't see in **Screenshot 6** is under the global tab:



SCREENSHOT 5. We need to add the socket library to our project. It is as easy as browsing the library folder, selecting the sock directory, and clicking on the *sock.tbh* and *sock.tbs* files.



SCREENSHOT 6. This is a shot of the whole shooting match. It's a little bit of Visual Basic, a little bit of C, and little bit of MPLAB X all wrapped up as Tibbo Basic.

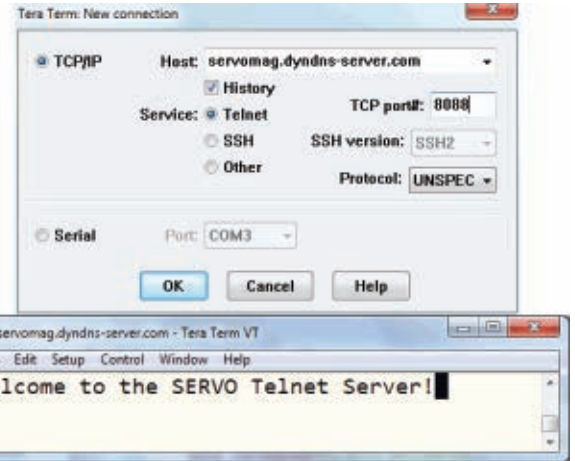
```
include "sock\trunk\sock.tbh"
```

Our Tibbo Basic networking application begins by including the *global.tbh* file, which is atop of every Tibbo Basic program:

```
include "global.tbh"
```

```
dim target_sock as byte 'assigned socket number
dim bitesIn as string 'incoming bytes to the
                        'socket
```

There should be no question about the *dim* statements. This is Basic, no matter where you find it. We are reserving some space for our socket number and the incoming socket data. The *on_sys_init* function is always invoked first and is used to initialize variables and setup methods. Our first task



SCREENSHOT 7. This is the result of dialing up the EM500EV over the Internet and pressing the ESC key.

is providing eye candy in the form of blinking the EM500EV's green LED five times:

```
sub on_sys_init()
```

```
pat.play("G-G-G-G-G-",1)
```

Following the light show, we head off to do some real work. Recall that we are using sockets. With that, we need to set up the IP portion of our local socket. These IP addresses belong to the EM500EV:

```
net.ip = "192.168.0.99"
net.gatewayip = "192.168.0.1"
net.netmask = "255.255.255.0"
```

To make the IP stuff work, we have to do some things at the gateway. I configured a socket using IP address 192.168.0.99 with a corresponding port number of 8088. Written on paper, this socket is described as 192.168.0.99:8088. The port portion of the socket is defined in the sock method:

```
target_sock=sock_get("")
sock.num= target_sock
sock.protocol=PL_SOCKET_PROTOCOL_TCP
sock.inconmode=PL_SOCKET_INCONMODE_ANY_IP_ANY_PORT
sock.reconmode=PL_SOCKET_RECONMODE_3
sock.txbufreq(2)
sock.txbufreq(2)
sock.localportlist="8088"
sys.buffalloc
```

```
end sub
```

The sock signature is a descriptive string you can put between the quotation marks of the *sock_get* method. It is permissible to leave the signature blank if the signature length is equal to zero. Trust me. It is.

The socket we are handed will use TCP protocol (PL_SOCKET_PROTOCOL_TCP). Our socket will also allow any IP address and any port address to connect (PL_SOCKET_INCONMODE_ANY_IP_ANY_PORT). Anything

goes connection wise on the UDP side of things, as well (PL_SOCK_RECONMODE_3). We have been generous with buffer space, allocating 512 bytes for the transmit buffer and receive buffer.

Just for giggles, we blink the green LED continually:

```
sub on_pat()
pat.play("~/G",1)
end sub
```

And, it's back to work:

```
sub on_sock_data_arrival()
if sock.num=target_sock then
  bitesIn = sock.getdata(255)
  if asc(bitesIn) = &h1B then
    sock.setdata("Welcome to the SERVO Telnet
Server!")
    sock.send
    pat.play("R-",1)
  end if
end if
end sub
```

A socket is a unique point at which data can enter an application. We could use our IP address of 192.168.0.99 to service a number of differing ports. Note the reference to a list in this statement:

```
sock.localportlist="8088"
```

Our application doesn't require any additional ports. So, when data arrives at the EM500EV's IP address, the *on_sock_data_arrival* event is fired. If our socket number was specified, we are entitled to retrieve the incoming data from the receive buffer.

In this example application, we are looking for the remote user to send an ESC character (0x1B). If an ESC character is received, we push out the welcome mat and blink the red LED. The Tera Term Pro windows used to make the call to the EM500EV are shown in composite **Screenshot 7**. I added a host to my DynDns account called *servomag.dyndns-server.com*. The *servomag* host entry points to the router in the EDTP shop. When all is said and done behind the scenes, the socket targeted by Tera Term Pro is 192.168.99:8088. Obviously, after clicking on OK in

Screenshot 7, I nailed the ESC key.

Lock the Door as You Leave

You will want to include the socket close and release methods in your production Tibbo Basic networking application. My job this month was to give you the keys to the Tibbo Basic warehouse. It's up to you to make sure you lock up when you leave. **SV**

Tibbo
Tibbo BASIC
EM500EV
EM500 "MiniMo"
BASIC-Programmable
Ethernet Module
www.tibbo.com

Saleae Logic
Logic Device
www.saleae.com

DynDns
www.dyndns.com

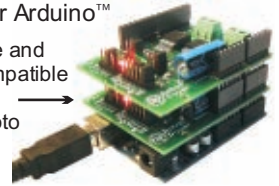


Extreme Robot Motor Control

High-Current motor control for Arduino™



Software and
plug compatible
with our
MegaMoto
shield



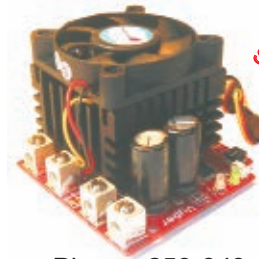
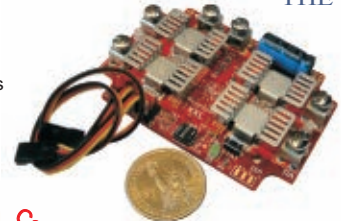
MEGAMOTO Plus Shield

- ◆ 30A Continuous with fan - 40A peak
- ◆ Current sensor outputs can be sent to any analog pin
- ◆ Stack up to three units on one Uno or compatible
- ◆ Independent half-bridges can control brushless/steppers too!



Scorpion XXL

- ◆ 7V - 28V
- ◆ Dual 40A+ Peak H-Bridges
- ◆ Current/Temp limiting
- ◆ R/C inputs w/mixing
- ◆ Optional enclosure
- ◆ 3.25" x 2.25" x .5"



Coming
Soon!

The Vyper

- ◆ Single 120A+ H-Bridge
- ◆ 8V - 42V
- ◆ R/C inputs w/mixing (2 units req.)
- ◆ Analog pot input
- ◆ Supports 8 AWG wires!
- ◆ Optional enclosure
- ◆ 2.85" x 2.25" x 1.75"

Phone: 253-843-2504 • sales@robotpower.com

FREE Stencil
with every prototype order

Embedded RFID
authenticate, track & protect
your product

www.magic-pcb.com

Call Tyler: 1 707 447 7744
sales@pcb-pool.com

PCB-POOL® is a registered trademark of **Beta**
LAYOUT
create:electronics

www.pcb-pool.com

3D Printers

This month:

by Michael Simpson

Part 1. Introduction into 3D Printers

Part 2. Assembly Highlights

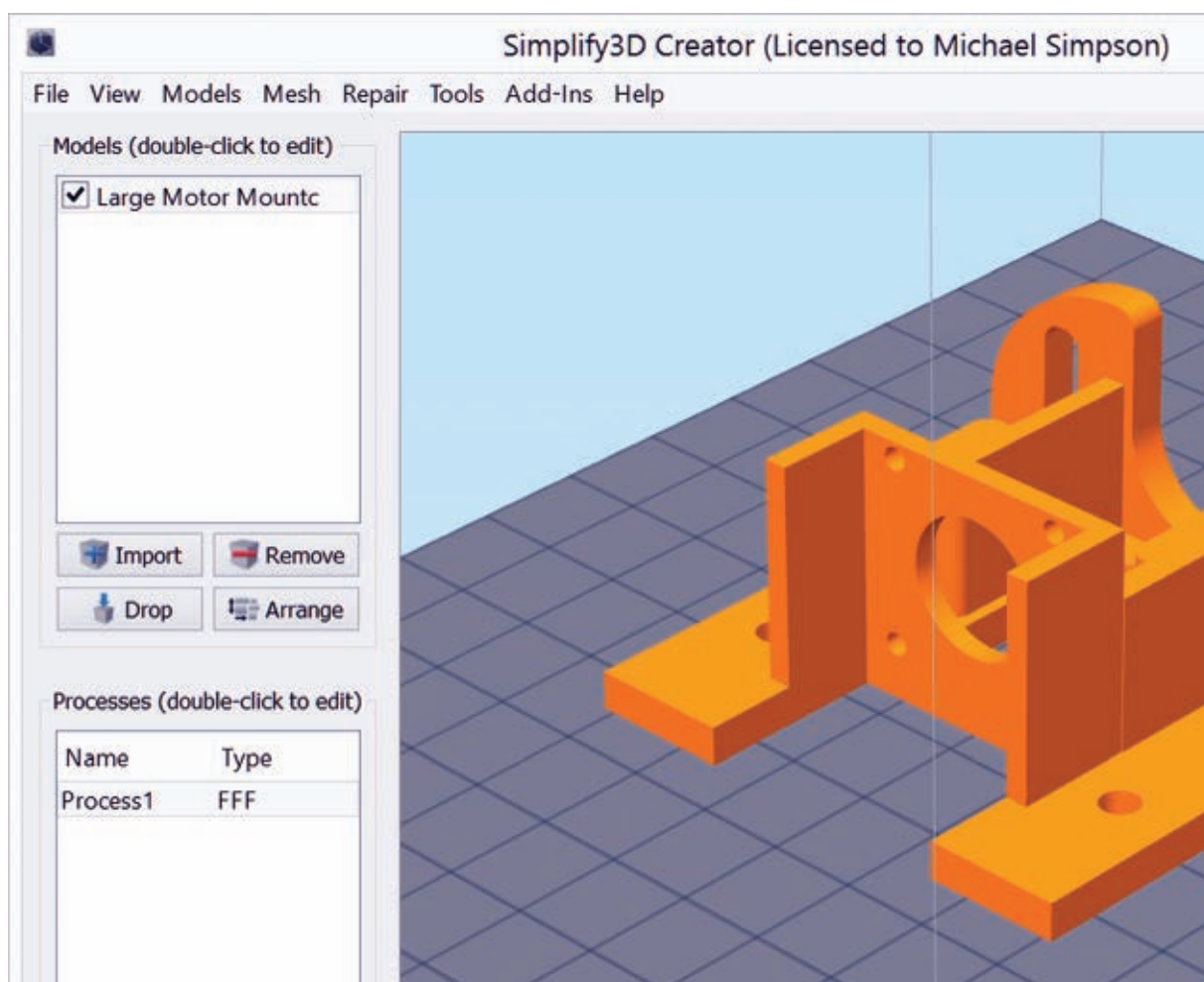
Part 3. Software and Configuration

Part 4. Tuning

Part 5. Upgrades

Part 6. Conclusion

Go to www.servomagazine.com/index.php?/magazine/article/august2013_Simpson for any additional files and/or downloads associated with this article. You can also discuss this topic at <http://forum.servomagazine.com>.



In this article, I will cover the slicer software used to convert your 3D models to instructions that your 3D printer can understand. I will primarily cover three software packages: Slic3r, Creator, and Afinia.

As an added bonus, my MakerGear M2 arrived since last time, so I will give you a quick peek at it before getting into the software.

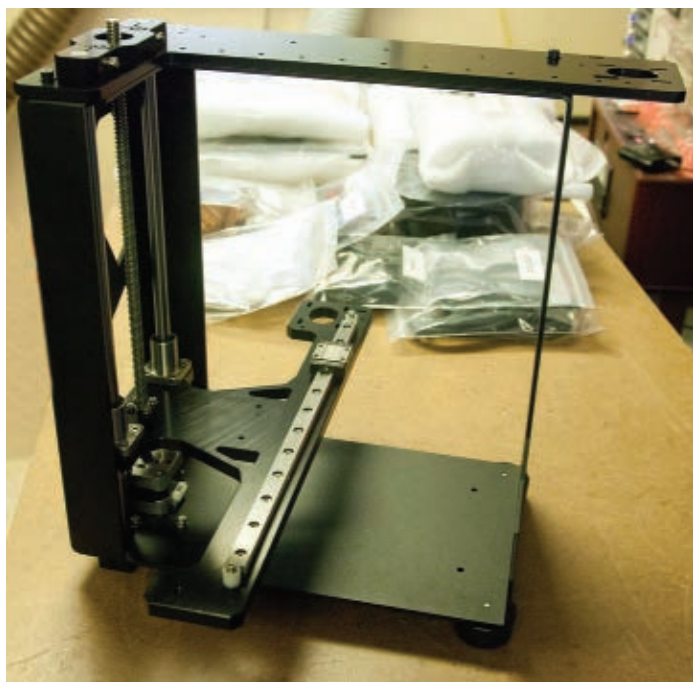


FIGURE 1.

MakerGear M2

I ordered the M2 in kit form and it arrived a little over six weeks later. The M2 comes partially assembled with all the remaining parts well labeled as you can see in the background of **Figure 1**.

Assembly was simple and took me a little over six hours to complete. The kit comes with most of the tools you will need for assembly.

The M2 is made primarily out of thick aluminum, and features linear rails on the X and Y axis. The Z axis utilizes two 10 mm steel rods with bearings and an ACME screw. Overall, the physical construction of the M2 (**Figure 2**) is the most rigid of any printer tested in this series.

The M2 comes with an 8" x 10" aluminum heated bed and a borosilicate glass platform shown in **Figure 3**. These are both removable, so you can use any 8x10 platform. I started experimenting with PLA and used both heated glass and acrylic with great success.

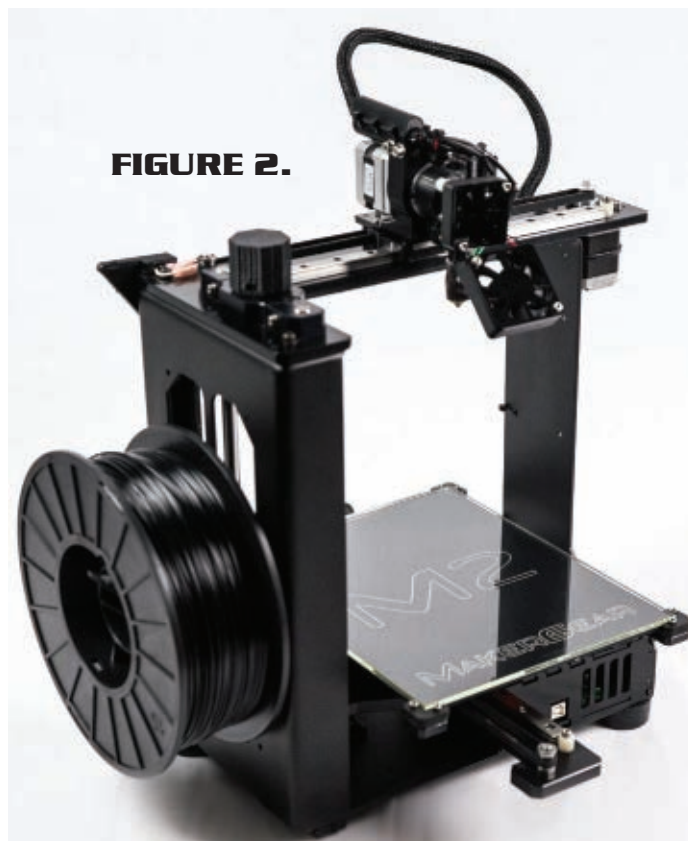


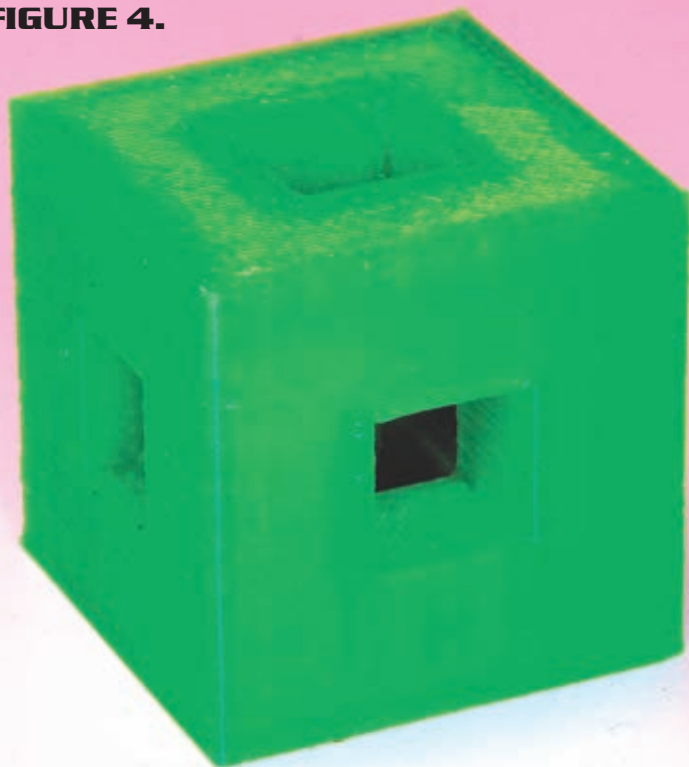
FIGURE 2.



FIGURE 3.

Please be sure to post any questions in the *SERVO Magazine* forums at <http://forum.servomagazine.com/viewtopic.php?f=49&t=16968>. I will also be posting additional information on my website at www.kronosrobotics.com/3d.

FIGURE 4.



How Does It Print?

I have had only a chance to print PLA, but it created quality prints with little or no tuning. The cube and Mr. Jaws prints (**Figures 4** and **5**) were the best prints I have achieved with any printer in this series. I'm so happy with this machine that I decided to utilize it for most of the prints used in the software section of this article.

3D Printing Software

It takes four pieces of software working together

to create a 3D print.

Slicer and CAD Software

The slicer software takes an STL file and converts it into instructions the 3D printer can understand. The STL file can be created by you, using the 3D CAD package of your choice. Many use Sketchup with an STL plug-in. All my work is done in Inventor or AutoCad. You can also download models from the 3D printer community via Thingiverse at www.thingiverse.com.

Controller Software

The controller software — also known as firmware — is what allows your 3D printer controller card to understand the slicer gcode commands. It's loaded directly onto your 3D printer controller board. You might think that this is something that you will not ever have to worry about, but that's not true. The 3D printer revolution is quickly evolving, so you may have to update the firmware as updates are made available. Updates are done through the latest version of the Arduino IDE.

Another reason for uploading new firmware is to tweak the settings for your controller card. I recently updated the Solidoodle controller card's firmware to allow me to extrude at a higher temperature.

Host Software

The host software is used to communicate with the 3D printer. It takes the processed gcode files created by the slicer software and sends them to the printer. The host software also acts as an interface to the slicer.

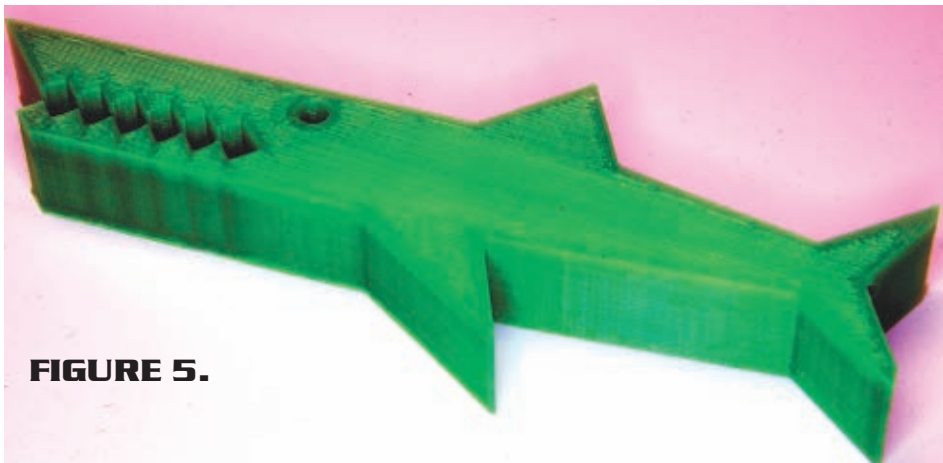
The host software gives you direct control of your printer. You can move the various axes and set the temperatures of the heated bed or extruder.

Slicer Software

While you may spend a lot of time using your favorite CAD software to model your 3D object, when it comes to printing, the bulk of your time will be spent setting up your host and slicer software. This is where the rubber meets the road when it comes to 3D printing.

I will be looking at many of the slicing parameters in Creator, Afinia, and Slic3r. Before I do, let's look at the packages.

FIGURE 5.



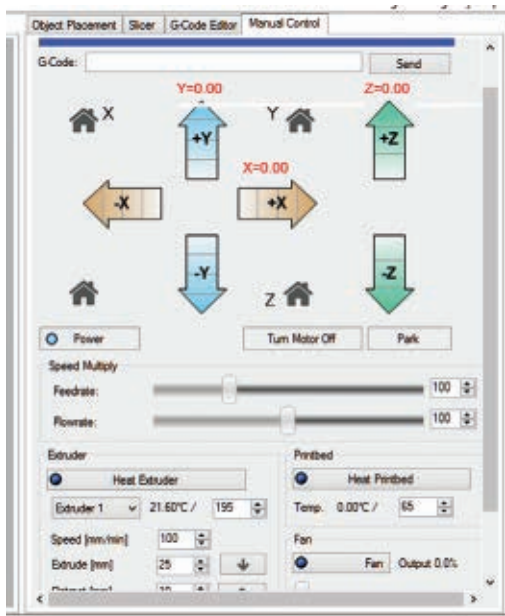


FIGURE 8.

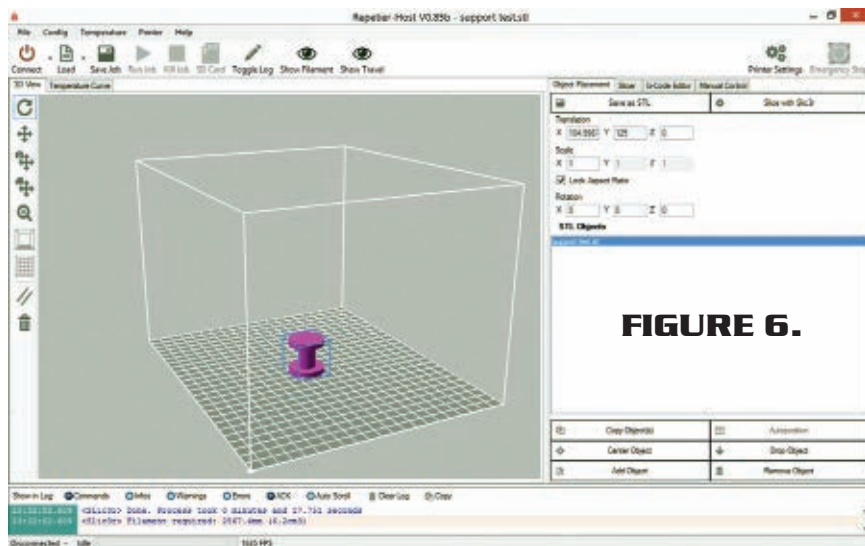


FIGURE 6.

Rostock Max

Both Repetier and Slic3r represent the open source and free end of 3D printer software. They both present an un-ending list of parameters that will help you fine-tune your machine. All of these parameters can also be overwhelming.

While there are other open source programs available, I have found these to work with most 3D printers. Since they are so popular, the settings for your printer should be readily available.

Repetier (shown in **Figure 6**) is your portal into the Slic3r software. Once the 3D object is loaded, you use Repetier to call up Slic3r to set your slicing parameters. Slic3r (shown in **Figure 7**) is where you will set all your slicing parameters covered in the next section of this article. Once set, you actually use Repetier to start the actual slicing operation.

Once sliced, you can review the layers, then use the printer controller (**Figure 8**) to prep your printer and send the sliced gcode commands to the printer. As Repetier sends the commands to the printer, the object is displayed as a virtual representation of the print.

The Repetier software can be found at www.repetier.com. Slic3r is included with the download.

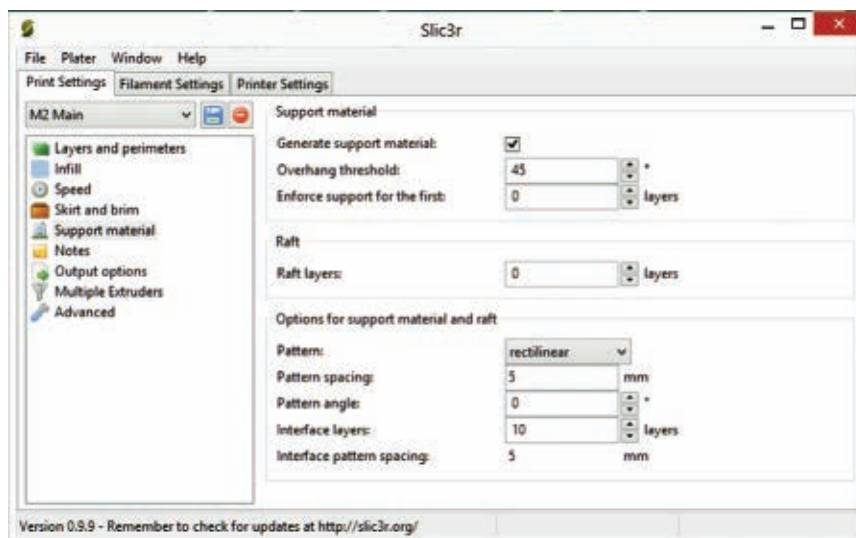


FIGURE 7.

Creator Host and Slicer

Creator from Simplifi3D is a commercial software package that sells for \$125. The folks at Simplify 3D created this package to make the software side of 3D printing somewhat easier. It's an integrated package that has some very advanced printing features.

One such feature that I have not seen in any other software is called "coast." It tells the printer how long to continue to print while it waits for the extruded filament to stop extruding plastic. This is particularly helpful when dealing with a less than perfect extruder, and will help you produce some remarkable prints.

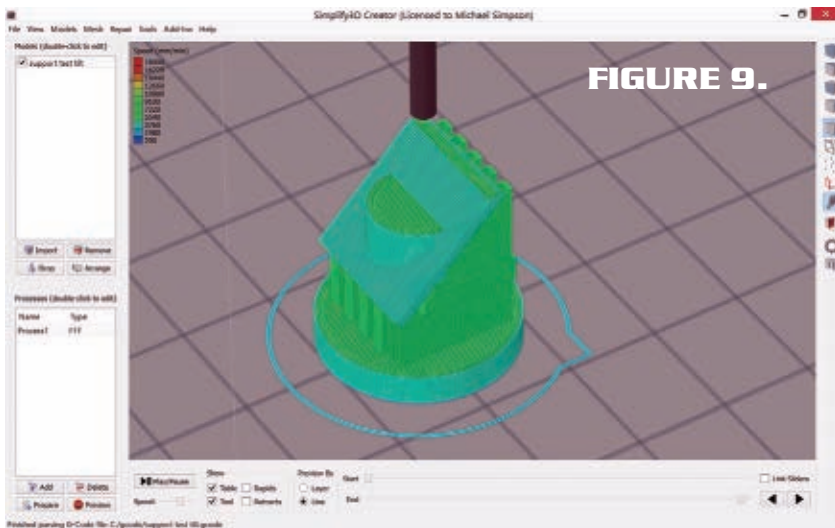


FIGURE 9.

Creator works much the same way as Repetier in the way you load your 3D object. Once loaded, you create a slicer process which will let you slice up your object for layer review, machine prep, and printing. The cool thing is that you can create as many processes as you want on your print. You can also tell each process to start and stop at various points in the print. This will allow a single print to use various layer thickness speeds and supports. (Refer to **Figures 9** and **10.**) Creator can be found at www.simplify3d.com/creator.

Afinia Host and Slicer

The Afinia software shown in **Figure 11** is proprietary and only works with the Afinia 3D printer. The Afinia software takes the simplistic approach to 3D printing. There are very few settings, yet the prints it produces are second to none. You load a 3D object much like the other packages, but that's where the similarities end. Most of the settings for the print are done in a single form as shown in **Figure 12**. As for controlling and prepping the printer, there is a small form with only a couple settings; most of the action takes place once you start the print. You can download the Afinia software at www.afinia.com/support/downloads.

Layer Printing Parameters

Now that you have seen the packages, let's take a look at the actual slicing parameters in detail. Almost all the parameters listed in this section apply to both the Repetier/Slic3r and Creator packages, but only a couple in limited form apply to the Afinia software.

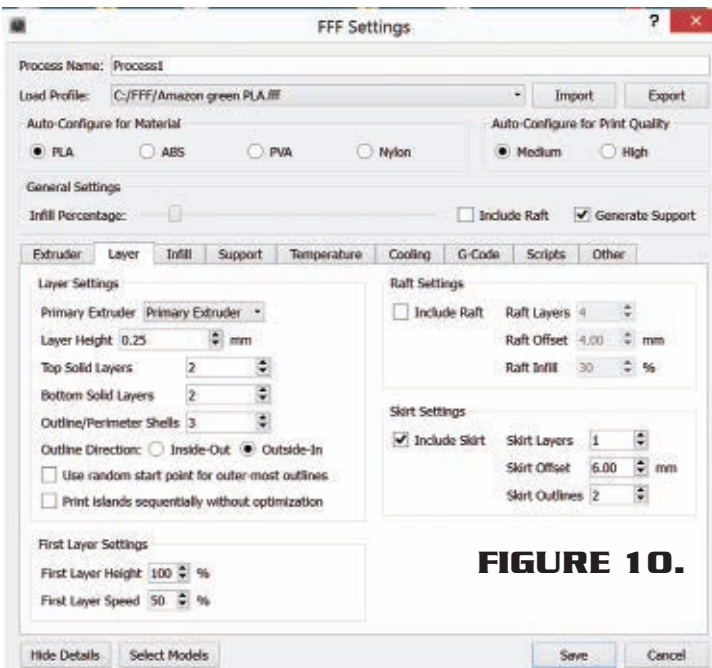


FIGURE 10.

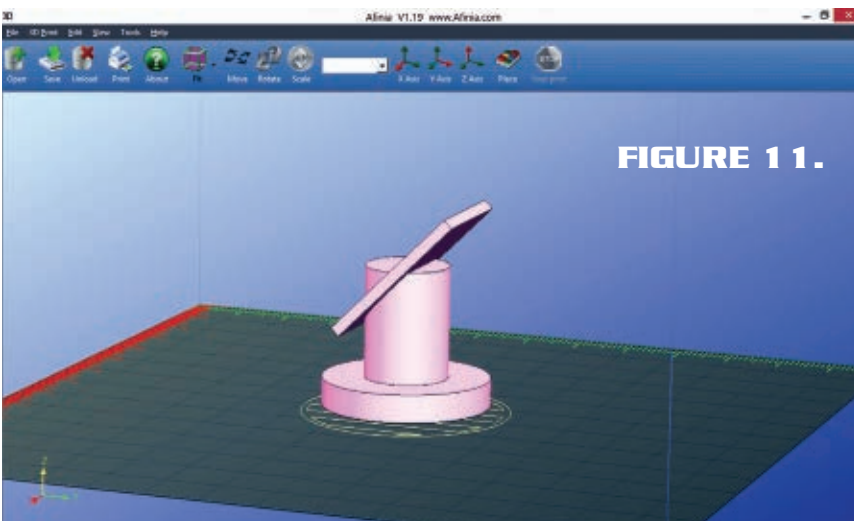


FIGURE 11.

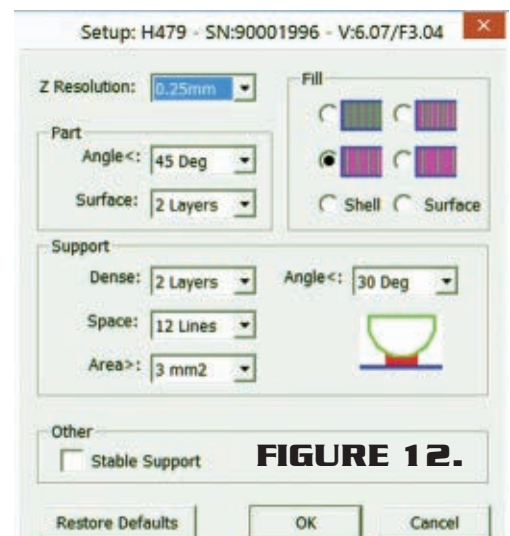


FIGURE 12.



FIGURE 13.

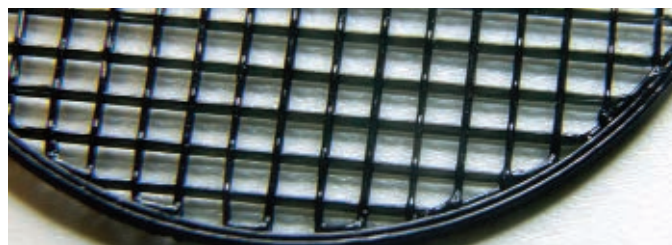
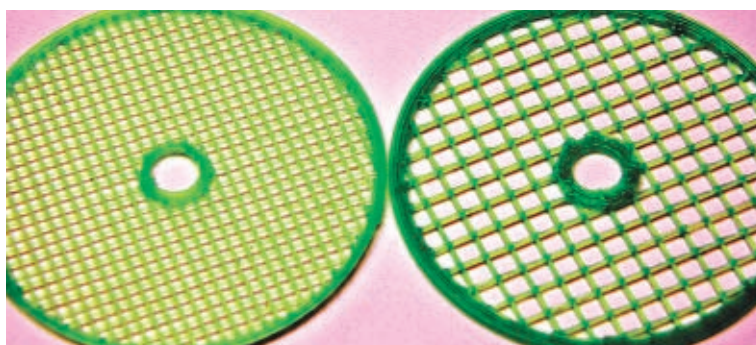


FIGURE 14.



FIGURE 15.

FIGURE 16.



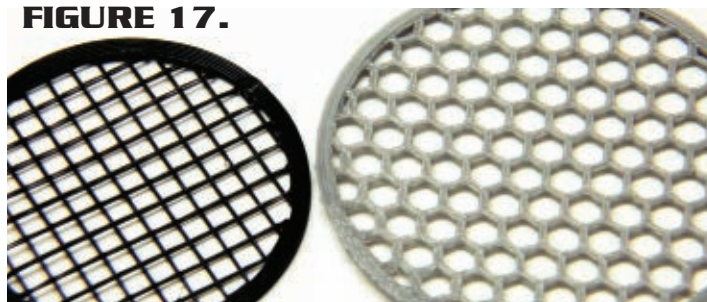
speed settings for these, as well. The Afinia software gives you no control over the perimeters.

Infill

When you create a 3D print, it is rare that the interior of the print is solid. You will set the percentage of infill which will dictate the amount of plastic that is used to fill the interior of the part. Referring to **Figure 16**, the print on the left has a 20% infill, while the print on the right has a 10% infill. In Slic3r, you set the infill percentage as a decimal number between 0 and 1.

In Creator, you use an actual percentage. The Afinia software does not give you granular control over the infill settings. You get four settings, ranging from almost solid to sparse. Selecting the amount of infill is a balance between weight, strength, and the amount of material you wish to use in your print. The more infill you use, the longer your print will take. Some slicers — like Slic3r — allow you to select the actual infill pattern. Referring to **Figure 17**, the pattern on the left is a rectangle which is supported by all slicers. The pattern on the right is honeycomb which is only supported by Slic3r.

FIGURE 17.



Layer Height

The slicer programs will have a setting called "Layer Height" on both the Slic3r and Creator software. On the Afinia, it's called "Z Resolution." This setting lets you control the thickness of each layer as it is printed. A setting of .1 mm to .3 mm is typical, but the slicer software will allow you to print thinner or thicker layers. The machine will dictate the actual range of reliable layer thicknesses.

The layer thickness lets you set a balance between resolution and print job speed. The thinner the layer, the more layers that need to be printed, thus the longer the print will take. Slic3r has an extra field called "First Layer Height" that allows you to override the layer height on the first layer. Creator has a more robust way of setting the height on any layer. Referring to **Figure 13**, the top print was printed at a layer height of .25 mm. The bottom print was printed at .10 mm.

Perimeters

Perimeters are the extrusions that go around the outside of the object you are printing. **Figure 14** shows a print that has two perimeters. The number of perimeters is determined by a setting in the slicing software.

In addition to the number of perimeters, you can set the speed to the outermost perimeters. Most of the time, you will be setting the outside perimeter slower than the others. The reason the software allows you to slow down the outer perimeter is to create the smoothest, most accurate profile possible. You can also have inside perimeters like the ones shown in **Figure 15**. Some slicers refer to these as small perimeters and give you separate

In most prints, you will have two or more solid layers on the top and bottom with your infill sandwiched between them. **Figure 18** shows the progression of the print as the printing takes place. The Slicer software will have a field for defining the number of top solid layers and bottom solid layers. A solid 40 mm x 5 mm cylinder is shown in **Figure 19**. Both the top and bottom solid layer counts have been set to zero. The perimeters have been set to one. These settings are typical for printing thin bracelets, tank treads, or even calibration objects.



FIGURE 18.

Skirts

A skirt is a perimeter that surrounds the main object. It

is used to give the extruder a chance to get its flow regulated before starting the actual object. The slicer software will allow you to set the number of loops (perimeters) and the distance from the main object.

Figure 20 shows a printed object with a skirt with three loops at a distance of 6 mm.

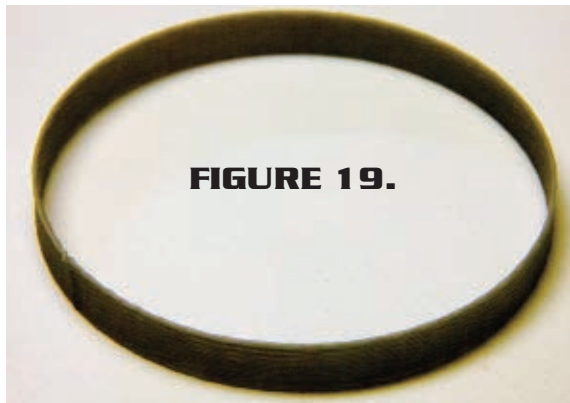


FIGURE 19.



FIGURE 20.

When a skirt is set to a distance of 0 mm away from the printed object, it is often called a brim (**Figure 21**). A brim is used to help the main object stick to the bed, and help with curling of the edges. Typically, a brim is only one layer thick to ensure it is easy to remove from the main object. While both Slic3r and Creator let you print skirts, Slic3r lets you print separate skirts and brims. You also have the option to set the height of the skirt. A high skirt can shield the main object from drafts while printing. The Afinia software does not support skirts in any form. It does have a sequence at the start of the print that cleans the nozzle.



FIGURE 21.

Rafts

A raft is used to help with bed adhesion and warping. In addition, they are often used to compensate for a build platform that is not perfectly flat or slightly out of level. The problem with creating rafts is that the raft may be difficult to remove from the bottom of the print.

The Slic3r software gives you little control over the size of the raft; this can result in a raft that is impossible to remove from the main object. Creator gives you the most control over the raft. The raft shown in **Figure 22** was

printed with Creator. The Afinia software only gives you the option of raft or no raft. To date, the best performing rafts I have printed have come from the Afinia.

Supports

While the current batch of filament-based 3D printers are remarkable in their ability to create accurate prints, they can't defy the laws of physics. Take the object in **Figure 23**; this object has overhangs that you cannot extrude without

the filament sagging or collapsing altogether. To successfully print this object, you will need to add supports. Both Slic3r and Creator can create support structures automatically, as does the Afinia software. Creator goes one step further by allowing you to add supports manually. The print shown in **Figure 24** shows the main object with the supports added. The support itself is very thin and its only purpose is to allow the extruded filament to form a bridge from support to support. In most cases, the support is one continuous structure that can be removed all at once. Slic3r lets you specify the distance between the support structures to affect support density. Creator lets you set an infill percentage to do the same. The Afinia software lets you specify the number of lines between each structure. In general, I have found the default support settings work best for most prints that require them. Also, keep in mind that in most cases you can print an unsupported overhang at 45 degrees or more.

Speed Settings

The speed settings for the three software packages are set up a little different. The Afinia has three settings: Fine; Normal; and Fast. In Creator, you set the default speed which applies to all operations. You have an outer perimeter override that is set to 70% by default, and a first layer override that is set to 50% by default. There is also a bridging speed override that is set to 100% by default.

Slic3r takes a totally different approach. You can set the speed for the following parameters: Perimeters; Small perimeters; External perimeters; Infill; Solid infill; Top solid infill; Support material; Bridges; Gap fill; and First layer speed modifier. In most cases, I set all the settings the



FIGURE 22.

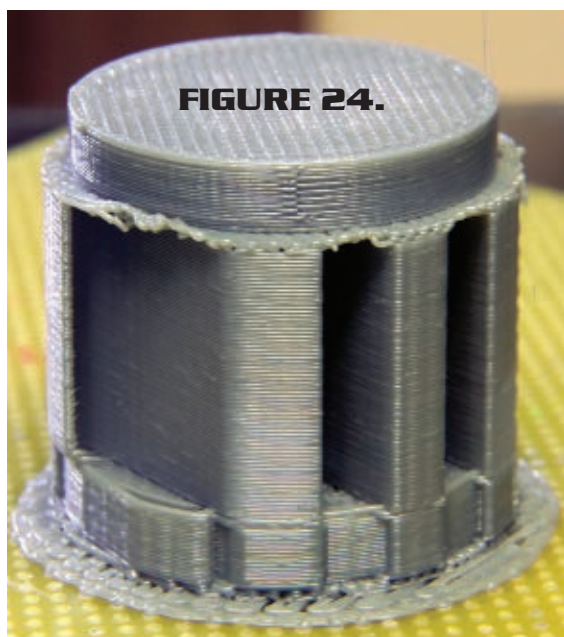


FIGURE 24.

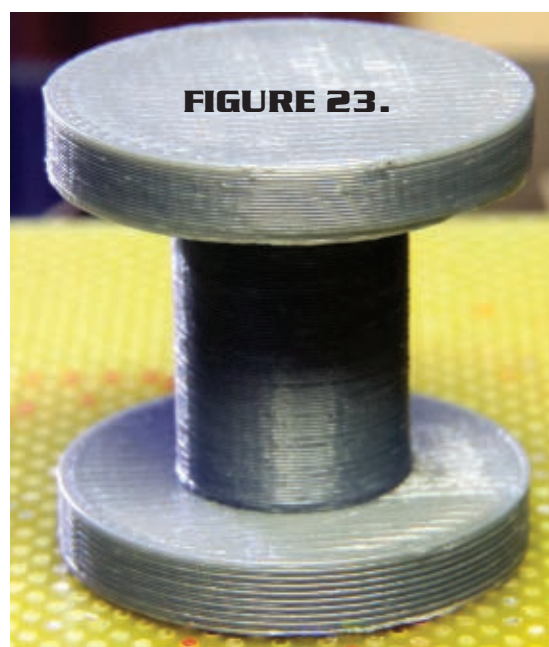


FIGURE 23.

same except for the first layer and external perimeters.

Your actual speed settings will be determined by your printer. The manufacturer should have the speed settings for your particular printer. However, you may want to experiment with a couple of them.

Final Thoughts

Slic3r and Creator have more settings available. Both have cooling settings that allow you to slow down the print or turn on external fans to help keep the printed part from overheating. There are script settings, offset settings, and filament settings. In addition, there are extruder settings for setting up and calibrating your extruder. These settings will be provided by the manufacturer, but feel free to tweak them as you see fit.

Next Month

I will take you through the process of calibrating your filament and tuning your machine. I will talk about the differences between printing ABS and PLA. I will talk a little about upgrades, and we will put some of the slicer settings to the test. **SV**

The *SERVO* Webstore

Attention Subscribers ask about your discount on prices marked with an *

CD-ROM SPECIALS



Free Shipping!

9 CD-ROMs & Hat Special

Only \$ 189.95
or \$24.95 each.

08.2013 FEATURED

New edition!

Includes a box of parts
containing more than
70 components.

Electronics
An Introduction
Hands-On Projects Edition

SALE
\$49.95!



ROBOTICS

Robot Builder's Bonanza, Fourth Edition

by Gordon McComb

Robot Builder's Bonanza, Fourth Edition includes step-by-step plans for a number of motorized platforms. The book is described as a compendium of robotics topics, containing more than 100 projects, including 10 robot designs new to the fourth edition. These modular robots are low cost, and are made to be reproduced by readers with no training in mechanical construction.

\$29.95*

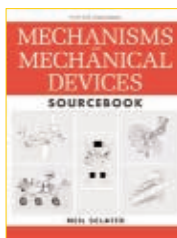


Mechanisms and Mechanical Devices Sourcebook 5th Edition

by Neil Sclater

Fully revised throughout, this abundantly illustrated reference describes proven mechanisms and mechanical devices. Each illustration represents a design concept that can easily be recycled for use in new or modified mechanical, electromechanical, or mechatronic products. Tutorials on the basics of mechanisms and motion control systems introduce you to those subjects or act as a refresher.

Reg \$89.95 Sale Price \$79.95



Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists

by Dustyn Roberts

In *Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists*, you'll learn how to successfully build moving mechanisms through non-technical explanations, examples, and do-it-yourself projects — from kinetic art installations to creative toys to energy-harvesting devices. Photographs, illustrations, screenshots, and images of 3D models are included for each project.

\$29.95*



Build Your Own Humanoid Robots

by Karl Williams

GREAT 'DROIDS, INDEED!

This unique guide to sophisticated robotics projects brings humanoid robot construction home to the hobbyist. Written by a well-known figure in the robotics community, *Build Your Own Humanoid Robots* provides step-by-step directions for six exciting projects, each costing less than \$300. Together, they form the essential ingredients for making your own humanoid robot.

\$24.95*



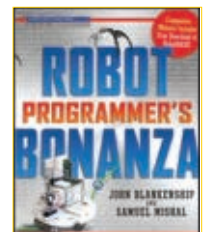
Robot Programmer's Bonanza

by
John Blankenship,
Samuel Mishal

**The first hands-on
programming guide
for today's robot
hobbyist!**

Get ready to reach into your programming toolbox and control a robot like never before! *Robot Programmer's Bonanza* is the one-stop guide for everyone from robot novices to advanced hobbyists who are ready to go beyond just building robots and start programming them to perform useful tasks.

\$29.95

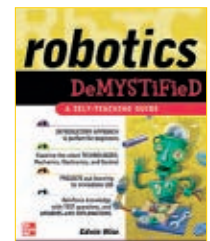


Robotics Demystified

by Edwin Wise

**YOU DON'T NEED ARTIFICIAL INTELLIGENCE
TO LEARN ROBOTICS!**

Now anyone with an interest in robotics can gain a deeper understanding — without formal training, unlimited time, or a genius IQ. In *Robotics Demystified*, expert robot builder and author Edwin Wise provides an effective and totally painless way to learn about the technologies used to build robots! **\$19.95**



**We accept VISA, MC, AMEX,
and DISCOVER**
Prices do not include shipping and
may be subject to change.

To order call 1-800-783-4624

SERVO Magazine Bundles



**Save \$10.00
Only \$57.95!**

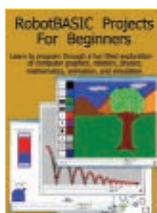
Now you can get one year's worth of all your favorite articles from *SERVO Magazine* in a convenient bundle of print copies. Available for years 04, 05, 06, 07, 08, 09, 10, 11 and 2012.

RobotBASIC Projects For Beginners

by John Blankenship,
Samuel Mishal

If you want to learn how to program, this is the book for you. Most texts on programming offer dry, boring examples that are difficult to follow. In this book, a wide variety of interesting and relevant subjects are explored using a problem-solving methodology that develops logical thinking skills while making learning fun. RobotBASIC is an easy-to-use computer language available for any Windows-based PC and is used throughout the text.

Price \$14.95

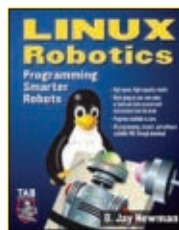


Linux Robotics

by D. Jay Newman

If you want your robot to have more brains than microcontrollers can deliver — if you want a truly intelligent, high-capability robot — everything you need is right here. *Linux Robotics* gives you step-by-step directions for "Zeppo," a super-smart, single-board-powered robot that can be built by any hobbyist. You also get complete instructions for incorporating Linux single boards into your own unique robotic designs. No programming experience is required. This book includes access to all the downloadable programs you need.

\$38.95



CNC Machining Handbook: Building, Programming, and Implementation

by Alan Overby

The *CNC Machining Handbook* describes the steps involved in building a CNC machine and successfully implementing it in a real world application. Helpful photos and illustrations are featured throughout. Whether you're a student, hobbyist, or business owner looking to move from a manual manufacturing process to the accuracy and repeatability of what CNC has to offer, you'll benefit from the in-depth information in this comprehensive resource.

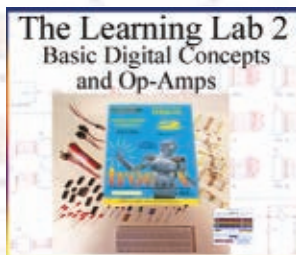
\$34.95



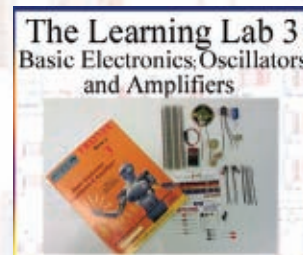
FOR BEGINNER BOT BUILDERS



\$59.95



\$49.95



\$39.95

The labs in this series — from GSS Tech Ed — show simple and interesting experiments and lessons, all done on a solderless circuit board. As you do each experiment, you learn how basic components work in a circuit, and continue to build your arsenal of knowledge with each successive experiment.

For more info and a promotional video, please visit our webstore.

Or order online
www.servomagazine.com

SPECIAL OFFERS

An Arduino Workshop
 Are you puzzled about the Arduino but finding it difficult to get all the pieces in the puzzle?
 Joe Pardue
 SmileyMicros.com

Puzzled by the Arduino?

Based on *Nuts & Volts* Smiley's Workshop, this set gives you all the pieces you need!

Book and Kit Combo \$124.95

Forbidden LEGO
 by Ulrik Pilegaard / Mike Dooley

Forbidden LEGO introduces you to the type of free-style building that LEGO's master builders do for fun in the back room. Using LEGO bricks in combination with common household materials (from rubber bands and glue to plastic spoons and ping-pong balls) along with some very unorthodox building techniques, you'll learn to create working models that LEGO would never endorse.

Reg \$24.95 Sale Price \$19.95

Beginner's Guide Complete Combo!

Combo Price \$229.95

VOL 3 Experiment Component Pack

"THE PHANTOM DRAW"

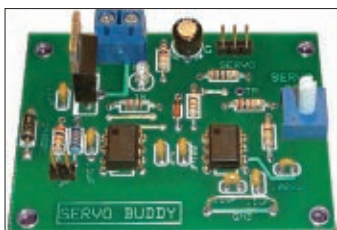
The KILL A WATT meter is the best way to help you determine your actual energy draw in ON and OFF home appliances.

Only \$29.95

To order call
 1 800 783-4624 or online www.servomagazine.com

PROJECTS

The SERVO Buddy Kit



An inexpensive circuit you can build to control a servo without a microcontroller.



For more information, please check out the May 2008 issue or go to the SERVO webstore.

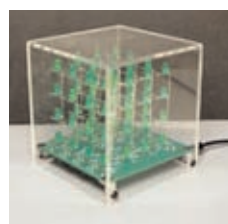
Includes an article reprint.

Subscriber's Price \$39.55

Non-Subscriber's Price \$43.95

3D LED Cube Kit

From the article "Build the 3D LED Matrix Cube" as seen in the August 2011 issue of *Nuts & Volts Magazine*.

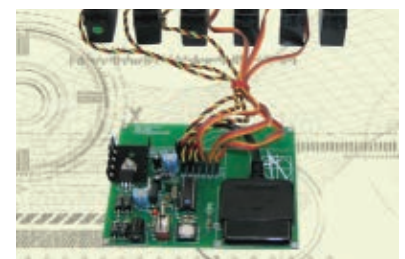


This kit shows you how to build a really cool 3D cube with a 4 x 4 x 4 monochromatic LED matrix which has a total of 64 LEDs. The preprogrammed microcontroller that includes 29 patterns that will automatically play with a runtime of approximately 6-1/2 minutes. Colors available: Green, Red, Yellow & Blue. Jig and plastic cases also available.

Subscriber's Price \$57.95

Non-Subscriber's Price \$59.95

PS2 Servomotor Controller Kit



This kit accompanied with your own PlayStation controller will allow you to control up to six servomotors.

Includes all components and instruction manual.



For more information, please see the February 2011 edition of SERVO Magazine. Assembled units available!

Subscriber's Price

\$79.95

Non-Subscriber's Price

\$84.95



Imagine it. Build it. Control it.™



SERVO ERECTOR SET V1.1



Introducing the Lynxmotion Servo Erector Set (SES) V1.1 with the new FlowBotics Studio V2. The SES V1.1 includes over 500 parts which are used to build and program a variety of different robots including a wheeled rover, biped, quadruped, hexapod and robotic arm. FlowBotics Studio is a powerful, versatile and easy to use graphical software suite which is compatible with a variety of robots and parts. FlowBotics Studio includes the complete FlowStone programming language!

Available now at www.Lynxmotion.com and www.RobotShop.com

Lynxmotion and FlowBotics Studio are trademarks of RobotShop Distribution Inc.



by Bryce Woolley and Evan Woolley

Tinker, Printer, Solder, Die

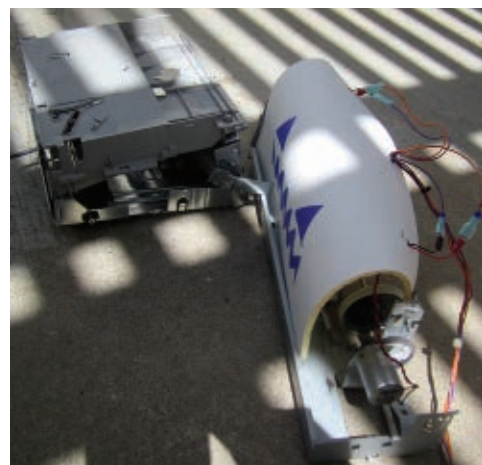
SERVO Magazine features a panoply of awesome projects, and every issue is sure to inspire intrepid tinkerers to pursue their own robotic endeavors. Sometimes, however, prospective hackers might be discouraged by what could seem like a lack of raw materials. Robots seem like a special breed of project, and if you don't have a kit of parts lying around or another robot that you're willing to cannibalize, it might appear insurmountably daunting to come up with the components needed for a cool project.

This month, we aim to show that anyone can make an awesome project out of just about anything. Since we're always on the lookout for spare parts, Robot Central has become a bit of a repository for old appliances and whatever else might be able to pay tribute to a future project.

To show that the possibility for robotic greatness lurks in even the most innocuous parts, we wanted to take a discarded appliance, turn it into a robot, and give it a task to do. Even better, we wanted to take two similar appliances, make two robots, and see which one could complete the task better. Let the games begin ...

The Honorable Schoolbot

A sense of friendly competition always makes things more fun, so we fashioned a set of rules to define the parameters of our game. Each competitor would start with a comparable appliance. We happened to have two printers laying around that we were sure had aspirations of greater things.



THE THRILL OF BATTLE.

The core principle of the game would be that to the extent reasonable, competitors would be limited to the parts in the appliance itself, but any part of the appliance would be fair game to use. The most strictly enforced aspects of this rule would be that only the motors from the appliance could be used, and no additional nuts or bolts could be added.

Perhaps it would be more efficient to list the exceptions to the rule of no additional parts that we would allow. Even though we would not add any nuts or bolts to the project, we couldn't resist allowing the introduction of some of the most universally useful essentials for any project — robotic or not: duct tape and WD-40. We like to add zip ties to make it a trifecta of awesome, so we would allow those too.

Another exception that we justified with the refrain of safety related to the electronics. We would use third-party battery packs instead of trying to power the robots off of the power supplies that came with the appliances and plugged into the wall. In our defense, the battery packs were also simply lying dormant in Robot Central, patiently waiting to be called forth for a



project just like this.

We aspired to limit ourselves to the wires that came with the appliance, but it soon became clear that this rule would be more trouble than it was worth. The printers only had very small gauge wires that we thought were not tough enough to handle very much current or physical strain, so we allowed extra beefier wires to be used. Perhaps our biggest concession was with the controls. Many appliances have knobs, buttons, and switches aplenty, so it might seem curious that we would want to make an outside hire here. Two major motivations shaped this decision.

The first (maybe sort of lame) reason was because a lot of appliances may come with buttons and switches and knobs (oh my!), but a lot of these components are surface-mount devices (SMDs). As you may know, SMDs have small poles and leads and are mounted directly into a printed circuit board (PCB). Small little leads meant for solder pads don't necessarily translate that well to a control box that would be more efficiently made from discrete parts.

Discrete parts are what you find inside of an older appliance — they are often larger components with bigger leads, where the wiring is done with actual wires instead of the printed paths on a PCB. In addition to convenience and efficiency, we also justified the use of outside components for the control box with the reasoning that this project could be merely the first round in a tournament of no-holds barred appliance death matches, and we would want a versatile control box that could be easily implemented with a range of projects.

With the ground rules in place, we were ready to start building!

The Constant Tinkerer

For our robotic grudge match, we thought that the most unassuming of appliances could have great potential as a mechanical gladiator — the humble printer. Printers seemed like

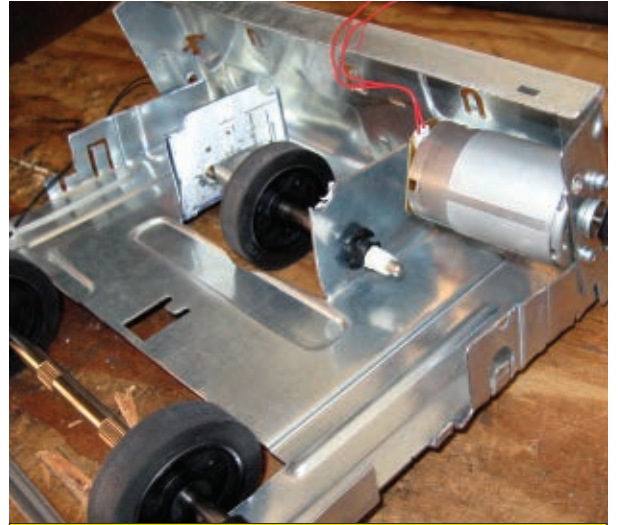
the perfect contenders for a handful of reasons (other than the fact that we happened to have two of them).

Firstly, since one of the major goals of this project — other than unfettered robot carnage — was to show that anyone can engage in the type of robot-building fun that populates the pages of *SERVO Magazine*, and old printers are exactly the type of appliance that a lot of people probably have laying around. You hesitate to throw it away because you know it has at least a few motors inside that could be used for something. (And here's something.)

Secondly, everybody knows that printers have to have some sort of locomotion to move the print heads around and feed the paper at least. Surely those motors could be put to better use destroying other printer robots.

The two printers we had were an HP Deskjet and a Dell. One of the great aspects of a robotics competition is the social aspect of testing your ideas against someone else's, so to bring as many brains into the project as possible we picked the Dell and our dad would take the lead at fashioning a robot out of the Deskjet.

With the printers doled out, we could start the project in earnest with the step that always looks like so much fun on HGTV — demolition. Deconstruction might be a more



LAYING OUT THE DRIVE TRAIN.



THE MOST UNASSUMING OF APPLIANCES.

appropriate term that better captures the challenge of this first step. Since



A FEROCIOUS BUT UNDERSTATED WEAPON.



OUCH! A DISMANTLED DELL.

recklessly tear open the case like a reckless child on Christmas morning, we didn't want to snap the best candidate for a floor piece or protective covering into bits.

As we took the printers apart, we got an idea of what we were working with and designs for combat robots began to swirl around our heads. Each printer contained three motors: one to translate the print heads from left to right across the

page, one to feed the paper, and a third vexing stepper motor. In the Dell, the stepper motor was used to move the scanner head. In the Deskjet, the stepper motor was used as part of a curious arrangement to move the ink cartridges.

Overall, we would say that the innards of the printers were pleasingly straightforward. Again, that's why we think projects like this are great — they can help to demystify the sometimes daunting world of robotics. Some things (not necessarily printers) can seem insurmountably complicated. However, once you steel your resolve, actually get started on a project, and take the time to slog through the basics, we're sure even the most timid tinkerer would find that they are perfectly capable of achieving robotic greatness.

Other than the motors needed for the aforementioned locomotive tasks, the insides of the printers were fairly sparse. There were some rollers to feed the paper, a few PCBs full of SMDs, the previously noted motors, and not too much else. One discovery that we made with respect to both printers is that while both were held together with tiny screws, neither seemed to have any nuts — all of the screws threaded into the printer frame itself.

With the printers apart, we set about the task of rebuilding them into ferocious fighting robots.

Smiley's Robot

The Deskjet printer our dad was

working with was an older model, and it was one that prominently featured a weighty metal frame. While the three motors seemed to promise the possibility of dedicating two motors to a driving base and a third to a weapon, that did not turn out to be the case.

The stepper motor lived up to its name and only gave a single step when powered. The stepper motors on both printers had small boards on them, so we thought that if they had some logic that could be the reason for the discrete steps.

Unfortunately, when we removed the board we found that it did not have any logic at all. The motor had four leads, and no combination of them achieved anything other than the small steps that were not really usable for a good weapon. So, with both printers, we were down to two motors.

For his basic design, our dad took inspiration from some simple R/C cars that were able to achieve a fair amount of maneuverability with a single motor drive. The drive train included one set of driven wheels and a third wheel on a pivot. When the robot reversed, the pivot wheel would move and cause the bot to reverse in an arc.

One of the metal frame pieces of the printer made the ideal floor — it came with numerous possible mounting holes, and the beveled surface and bent edges promised some decent strength. Just as with our FIRST robots and combat robots, the first order of business was to plan out the things we would attach to the floor because those decisions would dictate much of the rest of the design.

The main components that needed placing were the wheels and the motors. Figuring out what to use as wheels would be the first step because the Deskjet — like most printers — did not scoot around on a driving base. Our dad ended up pretty lucky with the Deskjet — the paper feeding mechanism included several large wheels perfect to use for the robot's drive train. Printers also conveniently contain several long metal



REINVENTING THE WHEEL.



MAKING SPARKS FLY.

we were largely limited to the parts in the printers themselves, we had a strong motivation not to break anything as we took the appliances apart. As thrilling as it would be to

shafts as part of the print head mechanism, and these made for ideal drive shafts.

With the wheels themselves sorted out, the next step was to figure out how to drive the motors. The printer came with a plethora of plastic gears, and many of them meshed nicely with the spindle on the motor. The two other motors (besides the unhelpful stepper motor) were simple DC motors that came with spindles on the shafts.

The smaller motor came with a spindle that meshed with many of the gears, while the larger motor meshed with a thin fine-toothed belt. The ideal shaft for the wheels actually came with a usable gear on one end, so the most efficient solution to drive the wheels would be to pick up that gear with the smaller motor.

To accommodate the wheels and to add some additional strength and protection, our dad cut some sections out of the floor panel and bent the front part of the panel upwards to create a front guard. The existing mounting holes for the motors were not conveniently located enough to be picked up for mounting the drive motor, but cutting off a part of the panel with the original mounting holes created the perfect template to locate the holes for a new mount.

After cutting a few more holes in the floor panel and cutting a slot for the pivot wheel, a simple drive train had taken shape. A quick test by connecting the motor leads to a battery pack confirmed that the Deskjet now had a mobile driving base.

The one motor drive may have been a simple and elegant solution for mobility, but it wasn't necessarily that dangerous. To give his bot a fighting chance (literally!), our dad resolved to add a spinning weapon to his robot using the other motor. The biggest challenge would be to devise a way to mate the motor to the weapon itself.

The motor spindle was designed for the thin belt, but tensioning the long belt to drive a spinning weapon seemed like a hopeless exercise in tedium. Instead, our dad devised a clamp using a section of the printer

frame with two threaded holes that looked to be perfectly spaced to hug the spindle.

After preparing the clamp, he discovered that the screws were a little too close together to fit over the spindle. As with every project, this obstacle was simply an opportunity for better design. He slotted the edges of the spindle with a rotary tool, and the spindle's new geometry gripped the spinning weapon tighter than it would have done otherwise.

With the quick addition of a protective covering using another panel from the printer held on using springs from the innards of the appliance, the Deskjet robot (let's call him George Smiley) was ready for battle. Now, all he needed was a worthy opponent.

The Bot Who Came In From the Cold

In many ways, the Dell printer we started with was very similar to the Deskjet that became George Smiley. It also contained three motors: two useful DC motors and a disappointing stepper motor. It came with an array of plastic gears, a few promising metal shafts, and a combination of plastic and metal frame pieces. We selected a long rectangular metal piece to serve as the main frame for the robot because it had several vertical projections that looked like possible mounting places for motors and things.

One of the big differences between the Dell and the Deskjet is that the Dell didn't have any large circular rollers that could be easily drafted into service as wheels. It was time to get creative.

The largest diameter circular things in the printer were a large plastic gear and a hollow metal shaft. The large gear — already at home on a shaft — would be a suitable wheel. The hollow shaft was appealing because it already came fitted with a plastic gear on one end that meshed nicely with the smaller DC motor.



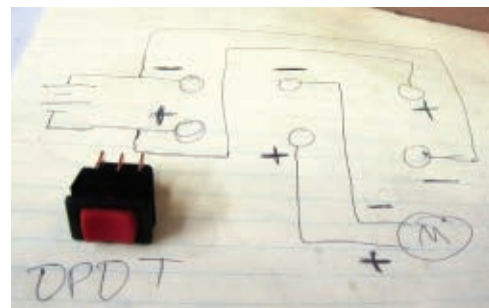
PRINTER'S PROGRESS.

Instead of struggling to find something to use as a wheel to put onto the shaft, we thought to go full-on Junkyard Wars and turn the shaft itself into its own wheel. We cut the shaft into numerous small cylinders, then encircled the remaining shaft with the cylinders. We used some epoxy to hold the assembly together. Epoxy was another exception to the printer-only principle because you know it's a good project when you mix up the epoxy components like an old-timey apothecary.

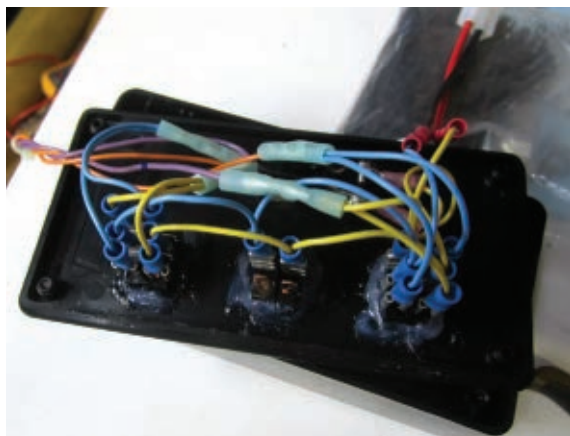
A rubberband held the cylinders in place and after it cured, we had



DUDE, YOU'RE GETTING A DRIVE TRAIN.



THE DPDT SWITCH.



WIRING UP THE CONTROL BOX.



GEORGE SMILEY AND MAGNUS PYM.

something that looked sort of like a sad imitation of an omni wheel. We now had something generally circular with a diameter approaching that of our other wheel, but it was still a little too small. We turned to another well settled exception to our printer-only principle — duct tape — to make up the difference.

Not only did the metal shaft have a suitable gear on the end of it, but one of the vertical supports on the frame piece we were using included the original mounting holes of that very shaft and the small DC motor originally intended to drive it. We thought we had already done enough reinventing of the wheel, so we picked up the existing mounts and had half the drive train done.

We resolved to make our printer bot more maneuverable than George Smiley, so we wanted to use the second DC motor to control the left drive wheel. We had the large gear to use as a wheel, and it fit nicely over a metal shaft. The solid metal shaft was a smaller diameter than the hollow one

and it actually fit inside of it, so we could have the end of the solid shaft sit inside the hollow one for support.

The solid shaft also came with numerous rubber stoppers around it, and they were the perfect bushings to make sure that everything stayed in place. The solid shaft had a gear on one end, but mating it to the motor would take some problem solving.

The spindle on the larger DC motor was meant for the thin fine-toothed belt. The teeth on the gear were helical, so no amount of squeezing the spindle and the gear together would get things to mesh.

As is often the case, the best solution seemed to be the most obvious one. The motor is meant to drive the

fine-toothed belt, so why not use the fine-toothed belt? We cut off a section of the belt, epoxied it around the plastic gear, and our problem was solved.

The belt was not perfectly straight on the gear, but the spindle was long and with sufficient force pressing the motor against the gear, we thought it would work just fine.

To mount the motor, we wanted to pick up the existing mounting holes in the metal frame, but the only problem is that the holes were in the floor instead of on one of the vertical walls. Fortunately, the placement of the holes were such that we were able to cut into the floor with our trusty rotary tool, create our own vertical support, and bend it into place to achieve the proper mesh with the gear. We used a piece of foam that came with the printer to create a rudimentary protective covering that we could doodle on with a marker.

The Dell printer was rechristened Mangus Pym, and was almost ready for battle.

The Printer War

With the drive trains and mechanisms done on the bots, the final task would be devising control boxes. We opted to use third-party switches for the control boxes for a few reasons.

One was for convenience, but the other was that Double Pole Double Throw (DPDT) switches would be the most advantageous for control, and the printers came with no such switches. With the DPDT switches, one pole could be for forward motion and the other could be for release; the stable center position meant that the bots did not need to be constantly on the move. We sketched out our diagram, placed the switches in the boxes, and wired everything up. For George Smiley's control box, we were actually able to repurpose the case of the printer's power supply.

With the controls finished, we were ready for battle. We constructed an enclosed arena with some two by fours and let the bots go at each other. Overall, we were pleasantly surprised by the decent maneuverability we were able to extract. While George Smiley's weapon created more noise than damage, it was pretty thrilling to have two fighting robots emerge from those most unassuming appliances.

After running down our battery packs, there was no knockout but George Smiley was judged the winner after scuffing up Mangus Pym's face.

We had a great time with this project. The overall goal was to demonstrate that anyone can experience the thrill and satisfaction of a robotics project even with the most innocuous of extra parts.

All of the principles and design strategies we used here are exactly the same as the ones we've used for combat robots, FIRST robots, and everything else. Don't let the lack of a kit discourage you from unleashing your inner robotics experimenter — the only limit is your imagination. **SV**

ServoCity

There was a time in the '60s and earlier when a prospective robot builder had only a few sources for parts. Hardware stores, junk yards, and surplus houses seemed to be the places early experimenters searched out, since there were no computer or robot stores in existence. Internet — what's that? Computers — you mean those big rooms filled with rows of cabinets with whirling tape drives? Electronic eyes — you mean photocells? Robot building is definitely a lot easier these days.

Building the mechanical aspects of a robot has not always been the easiest part of a robot project. I have written several articles over the years to steer prospective robot builders to various great sources of parts for their machines. Obviously, hardware, electronics, and computer stores (and certainly now the Internet with virtually millions of sources available at the touch of a few key strokes on your computer) are a great place to start.

Many first-time robot builders these days seem to have the most trouble with the mechanical aspects of their design. On the other hand, programming and using computers and microcontrollers — along with various software such as Linux-based operating systems, P-Basic, Spin, C, Willow Garage-developed Robot Operating System (or ROS), Microsoft Robotics Developer Studio 4, and others — are second nature to these same robot builders.

Many people interested in learning about robotics have started with robot kits such as the Parallax BoeBot and similar kits that range in price anywhere from less than \$50 to over a \$1,000. MINDS-i, LEGO, VEX, and similar kits can also assist a designer in basic prototyping.

Building a kit can help you learn the mechanical aspects and help you decide on just where your next design is heading. The more expensive Dongbu Hovis and Robotis Bioloid humanoid kits offer not only a very sophisticated bipedal robot, but the versatility for other designs as well.

After building several of the configurations offered in any of this range of kits, builders are ready to modify the designs to fit a particular

interest they might have. They find that their imaginations surpass the capabilities of the kits or the parts sources they have found.

When this happens, many builders would rather purchase individual parts for their specific design. Large hardware stores have many fasteners, structural beams, sheet metal, and plastic stock, but still are limited when the prospective robot needs hardware fittings designed for motion.

ServoCity

Enter ServoCity — a company whose products have appeared in many robots and other electromechanical products around the world. (This will actually be the first time I will feature a single manufacturer in my monthly column.)

ServoCity went about developing its product lines in a very systematic way. Back in 1994, Brian Thomas Pettey started the company in his dorm room at Southwestern College in Winfield, KS, and appropriately named it *Brian Thomas Robotics*. Initially, the company dealt in the manufacturing of educational robotic systems, but the name was changed to RobotZone in 2001 to encompass a larger realm of robotic products for commercial and government markets. These products ranged from toy robots and kits to more sophisticated products such as UAVs (unmanned aerial vehicles).

As founder and CEO, Pettey quickly saw the need and market for standard servo modifications to increase their torque or strengthen the bending moment of the output

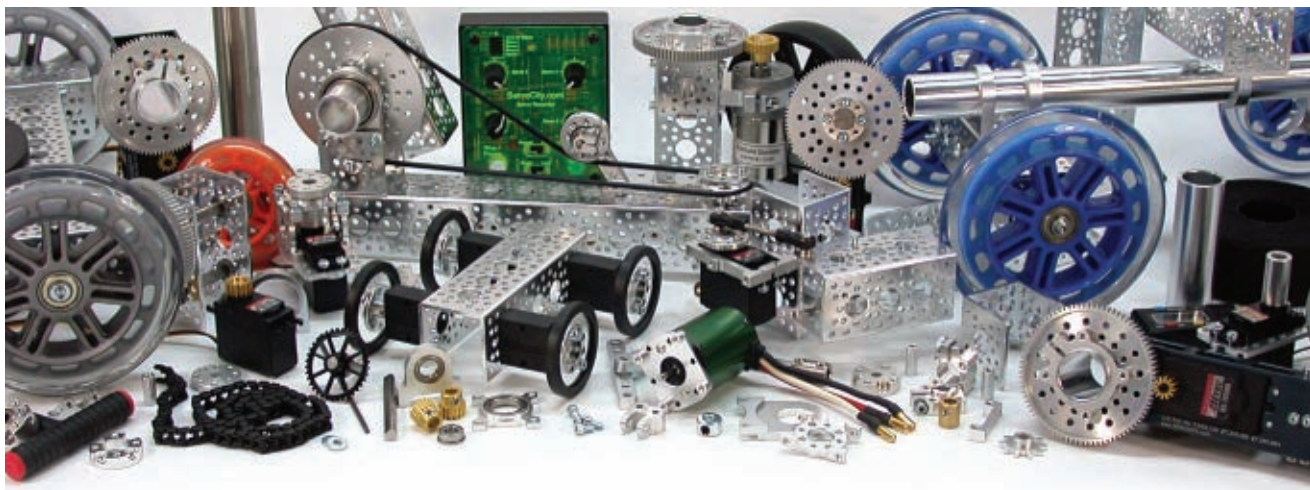


FIGURE 1. ServoCity products.

shafts for heavier robot designs. His company developed the RobotZone servo power gearboxes for higher torque (and proportionally lower speed), plus their ServoBlocks™ with a ball bearing pillow block-supported output shaft.

Today, RobotZone focuses on designing and manufacturing various products for the medical, industrial, military, and factory automation industries.

In the beginning, Petty realized that he was getting orders for parts that could be used to build pan-tilt-roll assemblies for film and video production that weren't available elsewhere. Seeing a need for quality — but inexpensive — pan-and-tilt platforms for small to large video and film camera systems, his line of videography accessories 'took off.' The same build-it-yourself attitude applied to his venture into robotic components.

According to Petty, "Instead of building full robots and trying to sell them online, we built the components so you could build your own to meet your specific needs. So, it's critical that all our parts work together."

Figure 1 shows just a few of the thousands of parts and components currently available for robot builders.

Today, the company focuses on developing ideas and designing products for individual hobbyists, universities, and fortune 500

companies. The most recent addition is their new line of aluminum mechanical components called Actobotics™ that can be found on their retail site (servocity.com).

ServoCity is RobotZone's online storefront which provides a large selection of exclusive parts for use in robotics, R/C, animatronics, videography, photography, industrial projects, and a wide array of other fields. As Petty states, "With nearly 20 years of experience in designing products for specific applications, they are a great resource for those who want to dream, design, and build!"

There are now roughly 6,000 other robotics components made or distributed through ServoCity and RobotZone.

Parts for Building Robots

It is easy to admire a new or improved robot that is on the market that can do something that no other bot can do. New robots can be much lower in cost and actually offer more features than their predecessor. It might have articulated hands and arms that can lightly hold an egg, yet be powerful enough to lift a hundred pounds from floor level.

These attributes and many other features might be breakthroughs in robotic technology, but have we even thought about that long and laborious

process that the company went through to develop their product? The same might apply to a home or lab built robot that has features that amaze us. How many dozens of prototypes did the designers, engineers, or hobbyists go through before the final design was completed?

Prototyping an Inexpensive Law Enforcement Robot

Builders might sit down and sketch out a robot arm and mobile base for use in small town police departments. Their number one question is typically determining the work envelope and joint forces required of the arm. They'll list any requirements such as a 10 pound payload, a gripper with two six inch curved fingers, a reach from the ground to 30 inches high, and a mobile base that can travel from very slow to 12 inches per second.

After determining a forward reach of 24 inches with the 10 pounds on the end, a decision is made to use linear actuators at the base of the arm and at the elbow to handle the forces needed.

An actual prototype arm is built and different types of joints, arm section lengths, and actuator mounting points are tried. The battery



FIGURE 3. Structural aluminum channel.

and motor controls are mounted in different locations to create a low center of gravity that is placed aft of the robot, but not too far back.

Figure 2 shows a mockup that the ServoCity R&D team made with Actobotics mechanical components to illustrate this very important process. The R&D team gave it to me to experiment with for this article.

Use Off-the-shelf Parts Rather Than Machining Your Own

The design process is certainly not this easy, but it can be made more effortless by having access to sensible and intelligently-designed mechanical components. Prototyping a particular electromechanical design can be a long and frustrating experience if the designer feels that all the components must be specifically designed for their final product.

If a designer is lucky enough to have a CNC mill or even a small machine shop at hand, making small mechanical parts can still be difficult. Taking a small part from the design requires careful analysis of the exact dimensions that are required and just how that part will fit with another part must be taken into consideration.

After the design is proven, it must be programmed into a CNC machine, or committed to a paper drawing for a machinist to machine by hand. Most small start-ups do not have any sort of machine shop or even a CNC mill.

Thomas Edison once stated back

in 1903, "Genius is one percent inspiration, 99 percent perspiration." You can dream up an excellent design or even put it down in CAD drawings or sketches, but the best mechanical designs are the result of putting "mettle" to the metal. The final design may get the glory but it is the 'guts' that give the electro-mechanical system its heart and soul.

The ServoCity Product Line

Let's review ServoCity's line of new products and illustrate how their intuitive interconnections make for easy construction of simple to very complex electro-mechanical systems. Of course, there is no way to cover

their complete line of products in this article, so I will cover the main categories instead.

I have found the mechanical components are intuitively designed to fit together in numerous ways and configurations. Their unique architecture makes them highly valuable for prototyping electromechanical mechanisms. The numerous beams, plates, flanges, and joint brackets make use of the thoughtfully designed hole patterns. These are not just rows of holes, but useful arrangements for sturdily mounting many different items.

Structural Members

Building a quick robot body



FIGURE 2. ServoCity wheeled actuator robot.

FIGURE 5.
Aluminum beams.

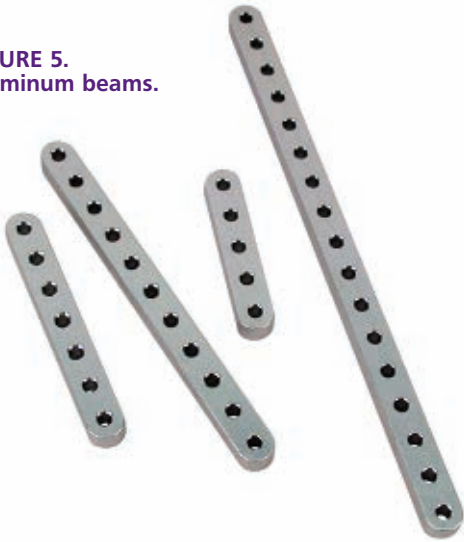


FIGURE 7. Delrin gear racks.

structure, arm, or legs for a prototype used to mean a trip to Home Depot or similar location for some angle aluminum or steel stock.

One of the things that stands out about ServoCity (in my opinion) are the very functional structural channels, beams, and tubes that robot and mechanical system builders can use to create numerous structures and arrays. No more hack sawing and filing stock to fit.

The 6061-T6 aluminum channels shown in **Figure 3** come in 10 lengths from 1.5" to 24" and are very robust, both from a bending moment point of view and from resistance to twisting.

Figure 4 shows an important attribute of these 1.5" x 1.5" external dimensions by 1.31" internal width channels that can hold large servos, as well as numerous fittings and bearings.

No more trying to drill precise holes spaced how you want them — these holes are sensibly spaced and a



FIGURE 4. Gearbox mounted in channel.



FIGURE 6. Stainless steel tubing.

builder can accurately mount numerous motors, servos, brackets, and other structural members to the channels.

Figure 5 shows the 1/4" x 3/8" aluminum beams from the Actobotics line that come in lengths of 1.54" to 7.70" with .38"/.77" hole spacings that fit most of their flanges, mounts, and other brackets.

Figure 6 shows an assortment of hollow 1" stainless steel tubing that comes in eight lengths from 2" to 12". Their line of 3/8", 1/2", 5/8", and 1" diameter aluminum tubing also comes in lengths of 2" to 12". These make excellent small robot arm sections, as well as structural mounting members.

Rotational and Motion Components

ServoCity's line of stainless steel solid shafting comes in 1/8", 3/16", 1/4", 5/16", 3/8", and 1/2" diameters, in one inch increments from 1" to 12" long. They also have 1/4" D shafts to 3" in length.

The shafting is not normally used as structural members but to transfer rotary motion from one location to

another. The hollow aluminum and steel tubing is also useful to transfer rotational movement from one point to another.

Bearings, bushings, shafts, gears, and similar motion products are not always easy to locate. Quite often, the items we find are not compatible with another item that we want to use in our robot project.

There might be differences such as metric vs. ASME threads to contend with. Shafts on one very nice gearmotor might not fit the gears that we want to use.

ServoCity has made this part of design easier with standard inch measurements. Most of the screws used with almost all of the brackets and mounting adapters are 6-32 — a very common screw size used here in the US.

Their 64 to 128 toothed aluminum hub gears with a 32 pitch are used on many robot designs, as well as the servo power gearboxes — several of which can be seen back on the **Figure 2** mobile robot arm. They also have nylon and Delrin gears of multiple pitches and mounting configurations.

They handle pinion gears for mounting on a servo, as well as driving the 12" Delrin gear racks shown in **Figure 7**. They carry aluminum and plastic sprockets for 1/4" chain.

Amazingly, ServoCity handles 43 different plastic sprockets from 16 to 100 tooth in two different tooth spacings, and five aluminum sprockets from 16 to 48 tooth, along with the steel and plastic chain to fit.

Wheels are an important part of any mobile robot and ServoCity handles many types. Their precision disc wheels for light robots come in four diameters from 2" to 5" and in 10 different colors. The black heavy-duty wheels shown on the robot base in **Figures 2** and **11** come in 4" and 6" sizes, whereas they also have some 3" and 5" colored wheels. Off-road tires and wheels and some miscellaneous types round out the line.

Making a Standard Servo Even Better

Two of Pettey's most popular product lines are the RobotZone servo power gearboxes and ServoBlocks. The use of a large standard servo such as the HiTec HS-7950TH with a 7:1 external gear ratio can deliver 3,402 oz-in or 17.7 foot-pounds of torque at 7.4 volts. (Now, that is a lot of torque for a small robot by anybody's opinion!)

The output for the SPG7950A-360 shown in **Figure 8** is 360° (actually 400°) of rotation, and uses an external pot attached to the output shaft for feedback. Yes, rotational speed is reduced to a bit less than 60° per second, but the result is a massive amount of torque.

ServoCity has 60 different models including a series of power gearboxes that fit in the above-mentioned channel stock. **Figure 9** shows three servos contained within the channels and two others to drive the base. There's a ServoBlock to rotate a claw (or end-effector) at the end of the arm assembly.

The ServoBlocks development was an excellent idea just waiting to happen. Many of us who have built smaller robots soon found that mounting a wheel on the servo's



FIGURE 8. An SPG7950A-360 power gearbox.

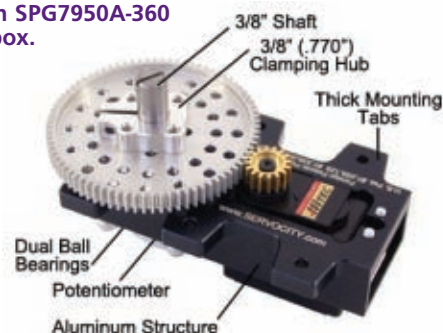


FIGURE 10. ServoBlocks with HiTec servo.

output shaft places a bending moment on the shaft. Most servos have plastic gears and a single small bushing on the output. Any bending of that shaft by the weight of the load on a small robot can destroy the servo.

Small robots such as the Boe-Bot can operate just fine with a small servo's output since they weigh only a few ounces. However, even a larger servo with metal gears and ball bearing output can be overloaded by bending when mounted on a larger robot's wheels.

The ServoBlocks shown in **Figure 10** add a second bearing so that there is no bending applied to the shaft's two bearings. Of course, there is still the torque loading to

consider on the internal gears, but for less than \$30 for one of the ServoBlocks kits to add to your metal-gear servo, good design practices will allow the servo to give long service.

Rounding out the rest of their line of components and parts are many fasteners, bearings, bushings, wires, and connectors. Their more complex products include the previously mentioned pan/roll/tilt camera platforms. ServoCity also has servos, joysticks, servo and motor controllers, and linear servos (like the ones in **Figures 2 and 11**), R/C radio gear,



FIGURE 11. Modified ServoCity wheeled actuator robot.



FIGURE 9. More ServoCity robotic components.

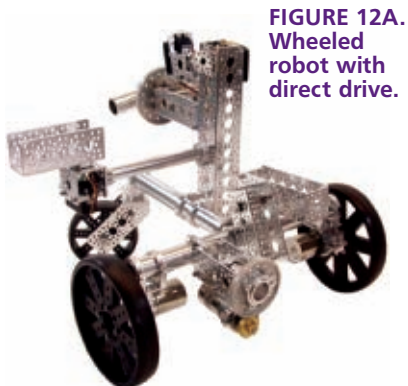


FIGURE 12A.
Wheeled
robot with
direct drive.

tools, and many more items of applicable interest to robotics and mechatronics. I highly recommend that you go to their site to view all their products. The site has dimensional drawings, clear photographs, photos of multiple uses, applicable mounting brackets, and accessories.

Building a Robotic Platform

As I mentioned earlier, the R&D people sent me the robot arm base in **Figure 2** to modify and use in experiments. They wanted to show the versatility of their various components and structural items, and get my feedback. I decided to modify it a bit with basic ServoCity (but also Actobotics) mechanical components,



FIGURE 12C. Chain drive configuration.

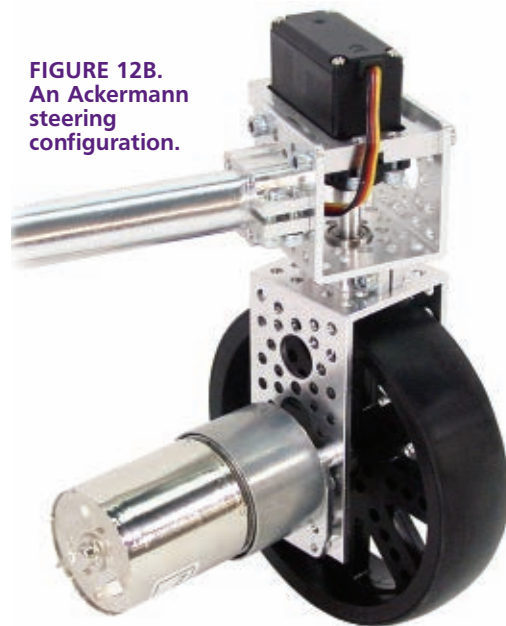


FIGURE 12B.
An Ackermann
steering
configuration.



FIGURE 13. Mars Rover for museum.

just to see how easily I could change the configuration using only these parts. **Figure 11** shows a few of the changes I made. I removed the chain drive and replaced it with two servos at the end. (Easy changing of configurations during a prototyping process is important in selecting a final design.) The two wheels are driven by a set of their 12 VDC gearmotors, of which they offer over 50 different models and gear ratios.

There are numerous ways to mount these gearmotors; **Figures 12A** through **12C** show a few of the ways. I used a miniature Firgelli L12-50-50-6R linear actuator instead of a standard servo for the gripper as it made for a more compact arrangement.

Petty and his team are customer-oriented and are very open to

suggestions from prospective robot builders. Their product line is constantly growing, and they aim to keep the robotics and mechatronics community supplied with as many parts and components as they can. Some have called ServoCity the 'Home Depot' of robot parts.

Robert Beatty and his two daughters built the Mars rover shown in **Figure 13** with ServoCity parts. The robot rover was built for a museum exhibit in New York City.

Final Thoughts

Though some of the ServoCity components and products might be a bit higher priced than you desire, you will certainly save a lot over trying to design and machine similar parts on your own. These components are very robust and intuitively designed.

Coupled with their gearmotors, servos, and motion products, a well thought out design can result in a great robot when using these parts.

SV

ROBO-LINKS

RobotShop.com



Pololu
Robotics & Electronics
WWW.POLOLU.COM




superbrightleds.com
COMPONENT LEDs • LED BULBS • LED ACCENT LIGHTS



Actobotics
DREAM. DESIGN. BUILD. REPEAT.
Precision building components from ServoCity.com



Ultrasonic Distance Sensing Made EZ www.maxbotix.com

USB-ProxSonar®-EZ™

- Great for kiosk applications
- USB interface
- Easy integration
- 1 inch resolution
- Calibrated beam pattern
- Starting at \$49.95

I2CXL-MaxSonar®-EZ™

- Incredible noise immunity
- I2C interface
- 1cm resolution
- Automatic calibration
- Great for UAV's and robotics
- Starting at \$39.95



www.canakit.com

- Over 200 Kits & Modules to choose from!
- Easy Worldwide Online Ordering
- Fast Delivery & Same Day Shipping

FM Transmitters • Timers • Audio Amplifiers • Motor Controllers



AndyMark
Inspiring Mobility
www.andymark.com



ALL ELECTRONICS CORPORATION
Electronic Parts & Supplies
Since 1967



THE ORIGINAL SINCE 1994
PCB-POOL®
Beta LAYOUT
WWW.PCB-POOL.COM

- Low Cost PCB prototypes
- Free laser SMT stencil with all Proto orders



SERVO 2009, 2010, 2011, 2012
Free Shipping!
9 CD-ROMs & Hat Special
Only \$ 189.95 or \$24.95 each.



INVEST in your BOT!



HiTEC
12115 Paine Street • Poway, CA 92064 • 858-748-6948 • www.hitecrod.com

For the finest in robots, parts, and services, go to www.servomagazine.com and click on Robo-Links.

ADVERTISER INDEX

All Electronics Corp.	18, 81	HiTec	2, 81	Robot Power	57
AndyMark	27, 81	Maxbotix	81	Servo City/Robot Zone	81, 83
AP Circuits	10	PanaVise	47	Solutions Cubed	7
AUVSI	19, 81	PCB Pool	57, 81	SuperBrightLEDs	81
Cana Kit Corp.	81, 82	Pololu Robotics & Electronics ...	3, 81	Univ. for Advancing Technology ..	39
EZSBC	10	Robotis	Back Cover	Vantec	47
Firgelli	51	RobotShop, Inc.	69, 81	Weirdstuff Warehouse	18
Global Specialties	15				

Motor Controllers • Arduino • Arduino Kits • Arduino Shields • Robot Platforms • Parts

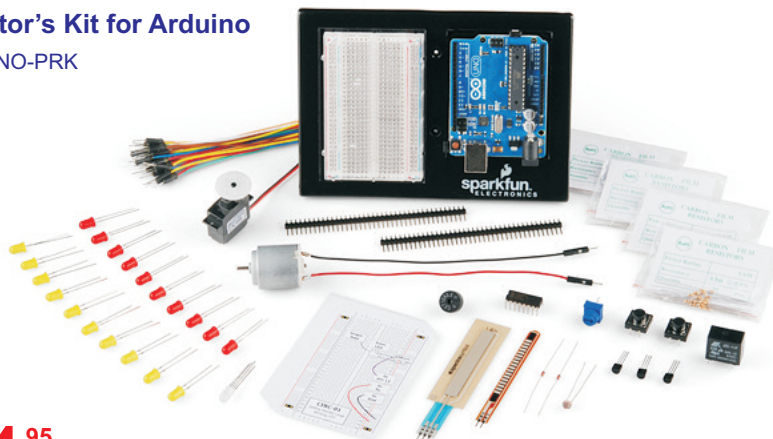
Arduino Uno



\$29^{.95}

ARDUINO-UNO
Latest "R3" Version!

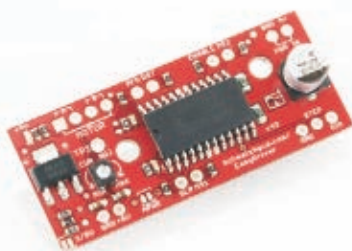
Inventor's Kit for Arduino
ARDUINO-PRK



\$94^{.95}

Includes 36-Page Printed Full-Color Manual

EasyDriver Stepper Motor Driver



SX09402

\$14^{.95}

Stepper Motor



SX09238

\$19^{.95}

50A Motor Speed Controller (PWM)
UK1133



Optional LCD

\$59^{.95}

**7A Bi-Directional
Motor Speed Controller (PWM)**



UK1125

\$39^{.95}

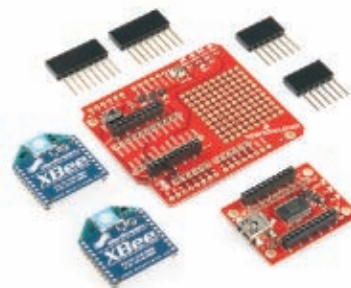
**USB PIC Programmer
(MPLAB Compatible)**



UK1300

\$69^{.95}

Xbee Wireless Kit



XBEE-KIT

\$94^{.95}

SERVOCITY

We have the parts for your ideas!

Shop Online | Call 620.221.0123 | sales@servocity.com

electronics



**2, 3, & 4
Function Joysticks**
starting at just \$19.99



**speed
controllers**



**Precision Digital
Speed Controller**
#ESC-72-24
\$99.99
- 10 to 14 VDC
- Up to 10A load
Precisely control gearmotors
and linear actuators!



**Digital Manual
Speed Controller**
#DMSC6-16-10
\$79.99
Proportional
Forward & Reverse!
- Linear control
- Built-in circuit breaker
- 6 to 16 VDC
- Up to 10A load



**Momentary &
Non-Momentary
Flip switches**
#RCE220 / #RCE210
\$9.99



Dual PWM Relay
(12 amp)
#RCE220
\$49.99



Single PWM Switch
(20 amp)
#EMOM1012
\$32.99



Dual PWM Switch
(4.5 amp)
#EMOM1022
\$34.99



Micro PWM Switch
(10 amp)
#RCE200H
\$34.99

PWM Switches & Relays



**Single PWM
Relay**
(10 amp)
#RCE210
\$39.99



**Dual PWM
Relay**
(24 amp)
#RCE225
\$59.99

servo controllers

**VERY
VERSATILE
UNIT!**



Dual Servo Driver
#902MSD
\$49.99



2 Servo Joystick
#JS200
\$99.99



4 Servo Recorder
#SPC2000
\$109.99



8 Servo Controller
#SC8460
\$69.99

NEW PRODUCTS



Hitec X1 Touch Charger
#44177
\$89.99



HS-900SGS Giant Scale
#34900S
\$424.99



GoPro Hero 3 Mount
#585518
\$9.99



H4 Smart Charger
#44186
\$274.99



**ITEMS
ADDED
WEEKLY!**

Brand New!

Actobotics
BALL-BEARING PRECISION BUILDING SYSTEM

**DREAM.
DESIGN.
BUILD.
REPEAT.**



...and much more!

Channel & Brackets Hub Mounts Bearing Mounts Gears & Sprockets Shafting & Tubing Hubs & Adaptors Wheels & Tires



**WORLDWIDE
SHIPPING!**

www.SERVOCITY.com

JOIN OUR ONLINE COMMUNITY!



High-performance networked actuators for robots

DYNAMIXEL

All in One-Actuator

High Power, High Precision

Full Modular Solution

Sophisticated Control Algorithms

Novel Gear Reduction System

Item	Unit	Data		
		H42-20-S300	H54-100-S500	H54-200-S500 (Preliminary Product)
Nominal voltage	V	24	24	24
No load speed	RPM	28.3	35.2	35
No load current	A	0.61	1.06	1.18
Continuous speed	RPM	15.59	32.7	32.1
Continuous torque	Nm	5.596	21.142	39.131
Continuous current	A	1.989	5.930	9.505
Resolution	Step/turn	304,000	502,000	502,000
Gear ratio	-	304	502	502
Backlash	arcmin	3.5	3.5	3.8
Interface	-	RS-485 / CAN	RS-485 / CAN	RS-485 / CAN
Operating temperature	°C	5~55	5~55	5~55



DYNAMIXEL PRO

Internal design structure (Left) / Actual product (Right).



Item	AX-12W	AX-12A	AX-18A	MX-28	RX-24F	RX-28	MX-64	RX-64	MX-106	EX-106+
Gear Ratio (material)	32 : 1 (enpla)	254 : 1 (enpla)	254 : 1 (enpla+metal)	193 : 1 (metal)	193 : 1 (metal)	193 : 1 (metal)	200 : 1 (metal)	200 : 1 (metal)	225 : 1 (metal)	184 : 1 (metal)
Network Interface	TTL	TTL	TTL	TTL / RS-485	RS-485	RS-485	TTL / RS-485	RS-485	TTL / RS-485	RS-485
Position Sensor (Resolution)	Potentiometer (300°/1024)	Potentiometer (300°/1024)	Potentiometer (300°/1024)	Contactless Absolute Encoder (360°/4096)	Potentiometer (300°/1024)	Potentiometer (300°/1024)	Contactless Absolute Encoder (360°/4096)	Potentiometer (300°/1024)	Contactless Absolute Encoder (360°/4096)	Magnetic Encoder (251°/4096)
Motor	Cored Motor	Cored Motor	Coreless Motor	Maxon Motor	Coreless Motor	Maxon Motor	Maxon Motor	Maxon Motor	Maxon Motor	Maxon Motor
Operation Voltage (V)	9.0 11.1 12.0	9.0~12.0	9.0~12.0	11.1 12.0 14.8	9.0~12.0	12.0~18.5	11.1 12.0 14.8	12.0~18.5	11.1 12.0 14.8	12.0~18.5
Stall Torque (N.m)	N/A	1.5 at 12.0V	1.8 at 12.0V	2.3 2.5 3.1	2.6 at 12.0V	2.5 at 14.8V	5.5 6.0 7.3	4.0 at 14.8V	8.0 8.4 10.0	8.0 at 14.8V
Stall Current (A)	1.1 1.3 1.4	1.5	2.2	1.3 1.4 1.7	2.4	1.5	3.9 4.1 5.2	2.1	4.8 5.2 6.3	6.1
No Load Speed (RPM)	360 430 470	59	97	50 55 67	126	67	58 63 78	49	41 45 55	69

USA E-mail : america@robotis.com / Tel : +1-949-333-3635

JAPAN E-mail : japan@robotis.com / Tel : +81-3-4330-3660

Etc. E-mail : contactus2@robotis.com / Tel : +82-70-8671-2609

DYNAMIXEL are registered trademarks of ROBOTIS Co., Ltd.

